ME 450
Winter 2022 Semester

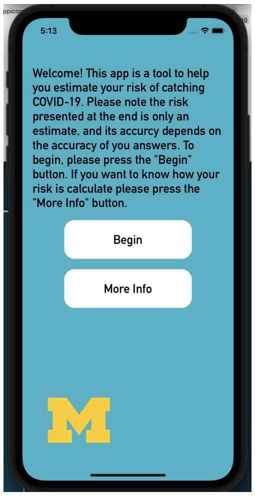
Real-Time Risk Assessment of Airborne Diseases

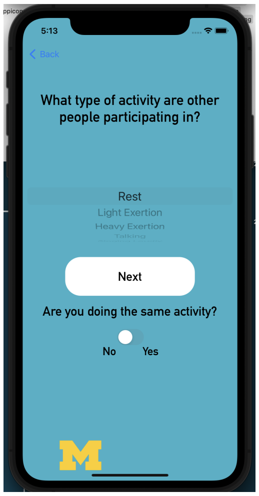
Final Report

Team 11

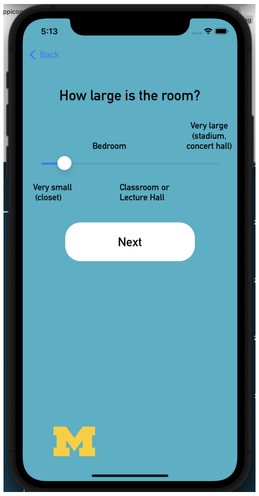
Robert Faktor
Andrew Hoelscher
Jacob Keener
Michael Scilken

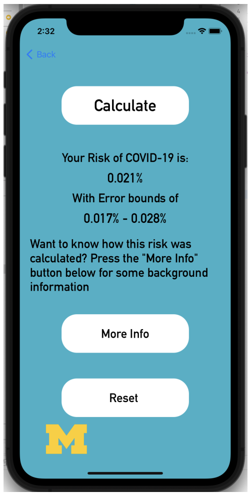
Professor Jesse Capecelatro, Project Sponsor
Professor Massoud Kaviany, Project Advisor


Welcome! This app is a tool to help you estimate your risk of catching COVID-19. Please note the risk presented at the end is only an estimate, and its accurcy depends on the accuracy of you answers. To begin, please press the "Begin" button. If you want to know how your risk is calculate please press the "More Info" button.


What type of activity are other people participating in?

Are you doing the same activity?


How large is the room?


Your Risk of COVID-19 is:
0.021%
With Error bounds of
0.017% - 0.028%

Want to know how this risk was calculated? Press the "More Info" button below for some background information

April 22, 2022

Final Report for the Real-Time Risk Assessment of Airborne Diseases

Robert Faktor, Andrew Hoelscher, Jacob Keener, Michael Scilken
ME 450 - WN 22 - Team 11

EXECUTIVE SUMMARY

Since its discovery in December 2019, SARS-CoV-2, also known as COVID-19, has caused major upheavals in global society. A significant hurdle facing individuals responding to this virus has been the difficulty of assessing the risks they face in certain environments. Although there are currently tools utilizing various epidemiological models to estimate risk, these are generally inaccessible to those without a scientific background, use some assumptions that are not always valid, and do not provide confidence intervals. Therefore, the goal of this project was to create a widely-available, easy-to-use smartphone app that can provide an accurate risk assessment of an individual's chance of infection of COVID-19 or other airborne diseases. This model improves upon existing tools by moving beyond the well-mixed assumption and by providing confidence intervals for the risk assessment, allowing users to know the strength of the estimate, and increasing accessibility by taking the form of an app.

The background information for this project is largely knowledge of epidemiological models that relate room parameters to the risk of infection. In particular, these models relate physical factors such as room size and ventilation rate to the concentration of viral particles, which is then related to the risk of infection. Additionally, a method for incorporating spatial variation in the viral concentration is discussed. The major challenges encountered include abstracting the more complex model parameters into intuitive user inputs, finalizing parameter values and relating them to qualitative inputs, and verifying the model using existing validated models.

The chief requirement for the app is accuracy. We also require that the app is easy to use and interpret in order to be widely accessible, and the calculations behind the app should be transparent. Finally, in order for the app to be convenient, the computation runtime must be fast.

Concept exploration was conducted to generate and select potential designs in the three following categories: user input questions, user interface designs, and the presentation of output results. For these categories, potential ideas were first developed using divergent thinking techniques, sifted through for feasibility, and were assessed using user surveys and tests. Qualitative user input questions and basic interfaces were selected

An alpha model has been developed and iterated in MATLAB with the main purpose of verifying the model programming and behavior. This alpha design was used to guide the beta design coded in Swift for iOS, and concepts selected through surveys, such as questions and interfaces, have been implemented into the beta design using Xcode. The alpha was used for internal testing for design verification, while the beta version was used in user testing for design verification. Sensitivity analysis has been conducted for the model, and model parameters have been researched and finalized for the model used by the app design.

The design was verified for meeting specifications for model accuracy, user input time, and computation time. However, the design has not been verified for user input accuracy, accurate interpretation of output, and transparent calculations. Therefore, the design can be improved, especially in areas such as the user interface. Finally, the uncertainty of the model could be developed to be more robust, possibly using a probability density function instead.

Final Report for the Real-Time Risk Assessment of Airborne Diseases

Robert Faktor, Andrew Hoelscher, Jacob Keener, Michael Scilken
ME 450 - WN 22 - Team 11

Abstract: *Current models of the risk of COVID-19 and other airborne infections often are inaccessible to those without a scientific background, rely on assumptions that are not always accurate, and do not provide confidence intervals. This project created a smartphone app design that is able to provide users with a means to determine their risk of infection of airborne diseases with confidence intervals for any room. This was accomplished by using a Dose-Response along with simple and qualitative user inputs (e.g., room geometry, ventilation, number of occupants) to output a probability of infection with confidence intervals.*

TABLE OF CONTENTS

INTRODUCTION………………………………………………………………………… p. 4
Problem Description…………………………………………………………………… p. 4
Design Contexts……………………………………………………………………….. p. 5
Stakeholder Context and Analysis…………………………………………………. p. 5
Intellectual Property Context……………………………………………………….. p. 8
Social Context………………………………………………………………………… p. 8
Environmental Context…………………………………………………………….. p. 8
Economic Context………………………………………………………………….. p. 9
Ethical Contexts……………………………………………………………………… p. 9
Design Process………………………………………………………………………… p. 9
Stakeholder Engagement…………………………………………………………... p. 9
Background and Information Sources………………………………………………. p. 10
Wells-Riley Model………………………………………………………………….... p. 10
Dose-Response Model……………………………………………………………… p. 10
Beyond Well-Mixed………………………………………………………………….. p. 13
Programming with Swift……………………………………………………………. p. 14
Information Sources………………………………………………………………… p. 14
REQUIREMENTS AND ENGINEERING SPECIFICATIONS………………………………. p. 14
Requirements………………………………………………………………………… p. 14
Specifications………………………………………………………………………... p. 16
Methods for Verifying Specifications……………………………………………... p. 16
CONCEPT EXPLORATION…………………………………………………………… p. 16
Concept Generation Methods……………………………………………………….. p. 16
Brainstorming…………………………...…………………………………….. p. 16
Mind Map……………………………………………………………………..… p. 16
Morphological Analysis…………………………………………………………….. p. 17
Concept Generation Results…………………………………………………………… p. 18
Concept Selection Process……………………………………………………………... p. 18
Alpha Design…………………………………………………………………………… p. 19
Validation…………………………………………………………………………… p. 23
Concept Analysis and Iteration…………………………………………………….. p. 24

TABLE OF CONTENTS

ENGINEERING ANALYSIS…… p. 24
    Sensitivity Analysis…… p. 24
    Model Reductions…… p. 26
FINAL DESIGN DESCRIPTION…… p. 27
FINALIZED PARAMETERS RANGES AND VALUES…… p. 31
DESCRIPTION OF VERIFICATION AND VALIDATION APPROACH…… p. 33
    Verification Testing Results and Future Plans…… p. 33
    Validation Testing Results and Future Plans…… p. 34
DISCUSSION…… p. 35
REFLECTION…… p. 37
    Design Contexts…… p. 37
    Inclusivity and Equity…… p. 38
    Ethics…… p. 40
RECOMMENDATIONS…… p. 40
CONCLUSIONS…… p. 41
ACKNOWLEDGEMENTS…… p. 43
NOMENCLATURE…… p. 44
APPENDICES…… p. 45
    Appendix A - Subset of Generated Concepts…… p. 45
    Appendix B - Default Model Parameters for Testing…… p. 51
    Appendix C - App UI Screenshots…… p. 52
    Appendix D - Model Parameter Uncertainty Values…… p. 54
    Appendix E - README for App Final Design…… p. 55
    Appendix F - Accuracy Verification Results…… p. 57
REFERENCES…… p. 58
TEAM BIOGRAPHIES…… p. 61

INTRODUCTION

Airborne viruses such as COVID-19, influenza, tuberculosis, and measles most effectively spread in indoor settings. Especially due to the ongoing COVID-19 pandemic, many individuals are concerned about the risk of airborne infection for themselves and others in varying environments. The project problem and background have been identified, analyzed, and researched. Through this, we have the goal of making an easy-to-use smartphone app that can provide an accurate assessment for risk of airborne infection. The design process is adopted from the ME 450 design framework, and the design contexts have been assessed. Requirements and engineering specifications have been developed through the design process while acknowledging the problem definition and design contexts. Concept exploration, including generation, selection, and analysis, has been conducted, and an alpha design was developed in MATLAB. Engineering analysis, including sensitivity analysis on the model parameters has been conducted. A beta version of the app has been developed in Swift. Verification testing has been conducted to assess the design against engineering specifications. Finally, the design process and project outcomes have been discussed and reflected upon, and recommendations have been made for possible improvements to the design.

Problem Description

The sponsor of this project is Professor Capecelatro. He has three main motivations for starting the project: the current models of the risk of COVID-19 and other airborne infections are generally inaccessible to those without a scientific background, they use assumptions that are not always valid, and they do not provide confidence intervals. This problem is important to solve because people currently have little concrete knowledge about the risk of infection they face in indoor environments. An app would help provide knowledge so people can make more informed decisions about their health [6]. Currently, the only risk assessment app on the App Store when searching "COVID risk," CoVis, estimates risk using questions about general activity, outbreak data for your location, and other personal health factors. Because of this, it cannot be tailored to estimate the risk of infection for a specific room or other indoor environment. Our app aims to accomplish this, because being able to estimate this kind of risk can be extremely helpful for people who want to make informed decisions about which activities are safe to take part in during the pandemic. Therefore, the goal for this project is to design and deliver an easy-to-use smartphone app that uses an accurate model to assess risk. The general idea of the app is described in Fig. 1 on p. 5.

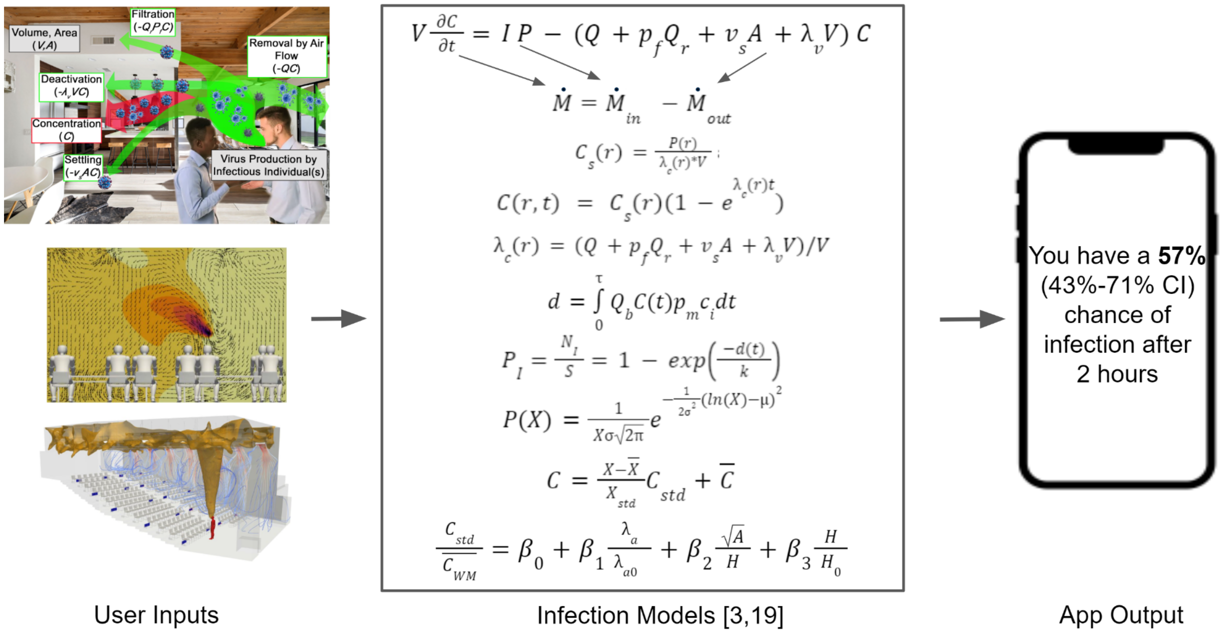


Figure 1: Flowchart of the general design goal of the project. The app will take user inputs about the room, such as height, area, and ventilation rate. These parameters will then be fed into the models that are described in the next section. Using these models, the app can present a clear probability of infection along with confidence intervals.

Design Contexts: In order to better understand and design a solution to the problem and need of this project, the broader contexts that have influence on our design must be explored and evaluated. As COVID-19 continues to create serious issues globally, the social impact of this project is the most significant context. Ethical contexts are also of high value to the overall design context.

Stakeholder Context and Analysis**:** A stakeholder ecosystem map, shown in Fig. 2 on p. 5 was developed to show the relationship between stakeholders, the roles they play in the project, and the potential impacts they could have on the solution. The different ecosystems include: resource providers (RP) shown in red, supporters and beneficiaries of the status quo (SB) shown in orange, complementary organizations and allies (CA) shown in green, beneficiaries and customers (BC) shown in light blue, opponents and problem makers (OP) shown in purple, and affected or influential bystanders (AB) shown in pink. The map is organized with concentric circles that show stakeholders's ability to influence the design solution. The innermost circle contains primary stakeholders who are directly affected by the problem and the development of a solution. The inner annulus contains secondary stakeholders who are part of the problem context, but may not be directly affected by the problem or the implementation of a solution. The outer annulus contains tertiary stakeholders who are outside the problem context but who still have the ability to influence the success or failure of the project.

Figure 2: Diagram showing the ecosystem stakeholder map for our project.

The five most significant stakeholder groups identified during stakeholder analysis were the project sponsor (Professor Capecelatro), the project advisor (Professor Kaviany), the sampled population through stakeholder engagement, the general target users of the app, and ourselves (ME450 WN22 Team 11). In order to visualize the relationships between different stakeholders, a simple and qualitative stakeholder map was iteratively developed, as seen in Fig. 3 on p. 7.

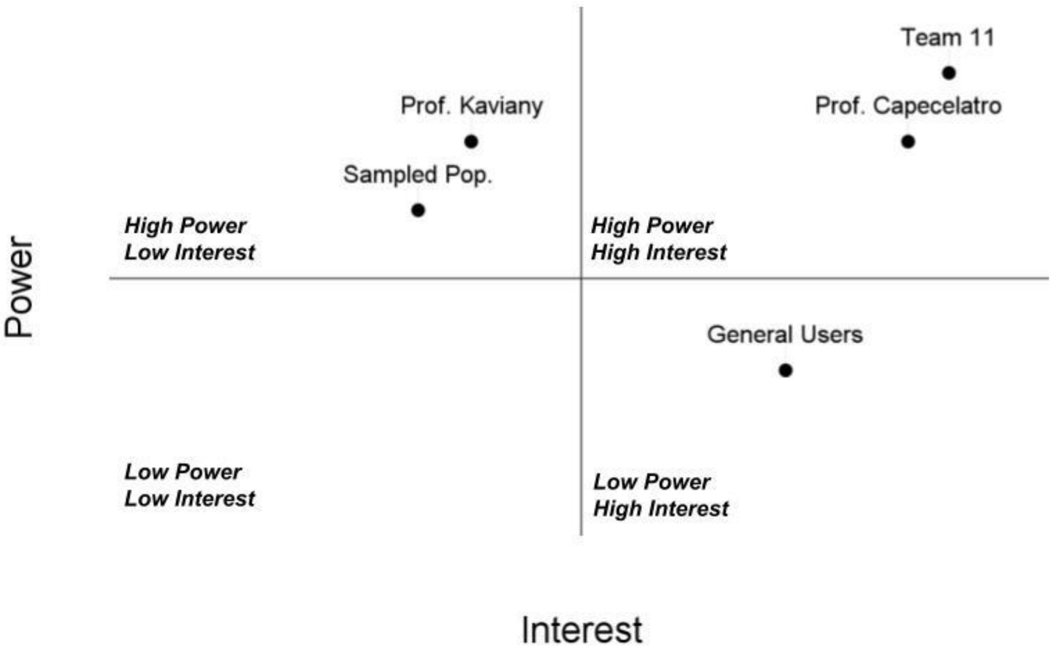


Figure 3: Stakeholder Map illustrating the power over the project design and interest in the project outcome held by different stakeholder groups, relative to one another

The project sponsor, Professor Capecelatro, was regarded as a stakeholder with both high power and interest. This project was a way to make the work Professor Capecelatro has been conducting towards improving airborne infection risk assessment models more accessible to lay people. Since the app uses models developed by Professor Capecelatro and his research team, he had both high interest and high power over the project.

The project advisor, Professor Kaviany, was regarded as a high power stakeholder with a lower relative interest. Professor Kaviany had significant power and influence over the project and our design primarily through his high engagement and meetings twice per week. However, his interest was lower because he has less personal involvement with the project background itself.

The sampled population was a stakeholder group with high power and lower interest. This population had high power because they were involved in surveys and testing as a part of our stakeholder engagement plan, and the results from this engagement directly influenced our design choices. However, these stakeholders were on average less interested in the project since they were asked to use the app as opposed to the general users who desire to use the app independently.

The general users are a stakeholder group with low power but high interest in the outcome of the project. This group is difficult to engage with, as it is a diverse and global population of anyone who is concerned with estimating their risk of airborne infections in any type of room. This is the group that the project is mainly aiming to serve. They have high interest because they seek the app out on their own.

Finally, as the engineers making the ultimate design choices, we (ME450 WN22 Team 11) are a stakeholder group with high power and high interest.

Additionally, the power dynamics between stakeholders should be considered, although no issues arose. It is important to acknowledge that compared to both Professor Capecelatro and Professor Kaviany, we are younger and less educated. Also, the general stakeholders group is constituted by many diverse lay people, so it is important to acknowledge that education, race, age, and more factors may impact the dynamics between the engineers and the users.

***Intellectual Property Context*:** Intellectual property (IP) has played little role in our project. There was no IP agreement made between any of the stakeholders for this project. Therefore, we held the IP to this project, and all work completed for this project did belong to the members of Team 11. The IP for this project is constituted by the files for the app that was designed as the solution for the problem of this project. The team has decided to give all of the completed work and the final app design to the project sponsor, Professor Capecelatro, for use at his own discretion.

Social Context: Social context, as a sustainability context, is the primary concern regarding the three sustainability contexts. The social contexts relevant to this project include public health concerns, quarantines, hospitalizations, and mortality rates associated with airborne diseases, which are all especially relevant due to the COVID-19 pandemic.

The societal aspects of the problem that is driving this project relates to the health concerns that people face. The contagious nature of COVID-19 and the potentially serious symptoms result in a necessity to gauge risk despite a growing percentage of the population becoming vaccinated against the virus. Currently, many people are still concerned with their or their loved ones' risk(s) of airborne infections, particularly from COVID-19. We hope the successful implementation of this app can help people make informed decisions in a given situation regarding infection risk. If this is the case, all stakeholders will benefit socially, especially the general users who wish to be aware of their risks.

Clearly, the spread of infectious disease is primarily a social issue from a sustainability perspective, and as a result, our project sponsor ranks social impact above all else. Additionally, providing those without technical backgrounds information about the risk of infection and the parameters that affect it is of high importance for both Team 11 and the project sponsor, Professor Capecelatro.

***Environmental Context*:** Environmental contexts of this project as a sustainability context are not of significant concern. This is mostly because the end goal of the project was to produce an app. There was no physical manufacturing, distribution, or disposal, and therefore none of the associated environmental costs. Instead, the environmental contexts include the mitigation strategies to reduce the risk of COVID-19 that may be increased to a change in perception of infection risk. Some of these mitigation strategies can be energy intensive, such as running HVAC systems in an effort to increase air changes per hour and decrease risk of infection. Additionally, waste management associated with increased mask usage and other preventative measures could increase as a result of this project in the case that users become more cautious after being aware of their infection risks. It is worth noting that although the end goal was an app, the necessary computational power and the energy associated with running the app is expected to be minimal.

Economic Context**:** Economic contexts as a sustainability context are not of significant concern. The smartphone app will be free and should have little to no development and upkeep costs. It is important to note that successful design and implementation could have an economic impact, as it could aid the world socially and economically in the return to a “new normal”.

Ethical Contexts**:** The main ethical context of this project was the concern of miscommunication regarding the risk of airborne infection. Our personal and professional ethics within our team align with those of the project sponsor, project advisor, and the University of Michigan. In this specific context, there were two main ethical issues. One ethical issue is that if we communicate a risk to the user in the app that is below the actual risk, the user may make potentially harmful decisions for themselves or others based on the inaccurate information. Also, another ethical issue is that if the app communicates a risk to the user that is higher than the actual risk, then we risk causing unnecessary stress and worry for individuals, and the users may even isolate themselves to avoid the higher perceived risk.

These ethical issues were finding the balance between using the most accurate room parameters as possible versus using app inputs that are simple to understand and estimate. This is essentially the balance between accuracy and accessibility of our app and model. The app is not an effective solution to the problem if it is not accurate in its risk generation, but is also not an effective solution if it is inaccessible. These values are in conflict because accuracy of the model depends on specificity of inputs, while more specific inputs might hinder accessibility by asking about information people do not know.

Design Process

The design process we followed is still the ME Capstone Design Process, which is described below. There are activities along each stage of the design process that we test our progress against, and there is room for iteration as new ideas and information are presented. This model was the most useful to us because it is a combination of stage-based and activity-based, which the course blocks teach to be the most efficient way to solve an engineering problem [22]. It also allowed us to focus on whatever part of the design process that we have the most information about, and it had the fluidity to allow us to weave our newest design changes into each step of our process.

The stages involved in the process are need identification, problem definition, concept exploration, solution development and verification, and realization. The activities that validate each step involves including relevant information from stakeholder engagement, mechanical engineering principles, ethical principles, and evidence-based decision making. All of the stages in the design process have been completed except for verification and validation and realization.

Stakeholder Engagement: Stakeholder engagement took multiple forms. We had interactions with our advisor and sponsor taking the form of weekly meetings with Professor Kaviany and meetings as needed with Professor Capecelatro. Additionally, we developed a survey that was sent to test users to learn about what the clearest and most efficient inputs are. This indicated that qualitative questions were the way to move forward with the app.

Background and Information Sources

Most of the prior work in this field is in the form of medical and scientific research that explores models of infectious diseases. Therefore, in this section we will provide a description of the primary models used in this field, as well as some of the tools that utilize them.

First, the models discussed in this report rely on the well-mixed assumption unless explicitly stated otherwise. This means they assume the concentration of COVID-19 particles does not depend on position and is instead uniform in space. This has been shown to not always be valid (meaning the concentration can change with location in many scenarios) [12, 17, 21]. It is used because it strongly simplifies modeling and allows for solutions that can be solved analytically.

Wells-Riley Model: A simple model of infection risk for airborne diseases is the Wells-Riley model, which is based on the idea of a *quantum* of infection. These quanta are defined as the minimum number of airborne infectious particles required to cause an infection. They are treated as discrete and randomly distributed in the air in a relatively low concentration, so the risk of infection from the quanta follows Eq. (1) [17, 18]:

$$P_I = \frac{N_I}{S} = 1 - exp\left(\frac{-I\, q\, Q_b\, t}{Q}\right) \tag{1}$$

where P_I is the probability of infection [% chance of infection], N_I is the number of people who will get infected [count], S is the number of people susceptible to getting infected [count], I is the number of infectors (sick people in enclosure) [count], q is the quanta generation rate [quanta/hr], Q_b is the pulmonary ventilation rate of a person [m^3/hr], t is the exposure time interval [s], and Q is the room ventilation rate [m^3/hr]. This is based on the well-mixed assumption, and is a simple mass balance relating the amount of virus particles exhaled by infected people in the room and the amount of virus taken out of the air by inhalation and room ventilation.

The key parameter here is q, the quanta generation rate. It is dependent on factors like the breathing rate of the infected person and whether or not they are wearing a mask. When using this model directly, q must be back-calculated based on data from existing outbreaks, where these factors are incorporated in the calculation of q. This means the model is not very versatile, as the values for q are only accurate in similar contexts.

Dose-Response Model: These parameters can be made more versatile using the dose-response model. Here, instead of using a pre-defined quanta generation rate, the concentration of virus particles C is calculated according to the following differential Eq. (2) [3]:

$$V\frac{dC}{dt} = I\,P - (Q + p_f Q_r + v_s A + \lambda_v V)\,C \tag{2}$$

where V is the volume of the room [m^3], P is the production rate [virions/hr], p_f is the probability of filtration, Q_r is the flow rate of recirculating air through a filtration system [m^3/hr], v_s is the droplet settling speed [mm/s], A is the floor area of the room [m^2], and λ_v is the deactivation rate of the virus [hr^{-1}]. This is simply a conservation of mass equation for the virus particles of the form $\dot{M} = \dot{M}_{in} - \dot{M}_{out}$, or more specifically: (Change in amount of virus) = (Virus produced by infectors I breathing out pathogens at rate P) - (Loss from ventilation, filtration, settling, and deactivation). This equation can be visualized in the following schematic seen in Fig. 4, showing the different methods of production and removal:

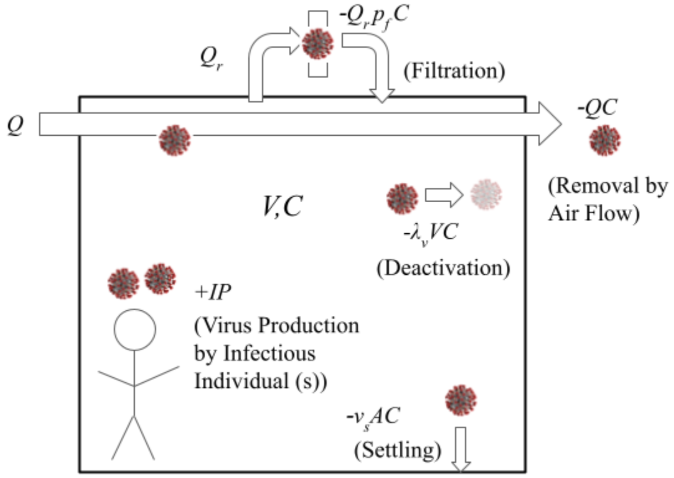


Figure 4: Diagram showing production and removal of virus from an enclosed room. The particles are assumed to be well mixed, so the removal methods are solely based on the concentration of virus in the air.

A more realistic version of the diagram seen in Fig. 4 can be seen below in Fig. 5.

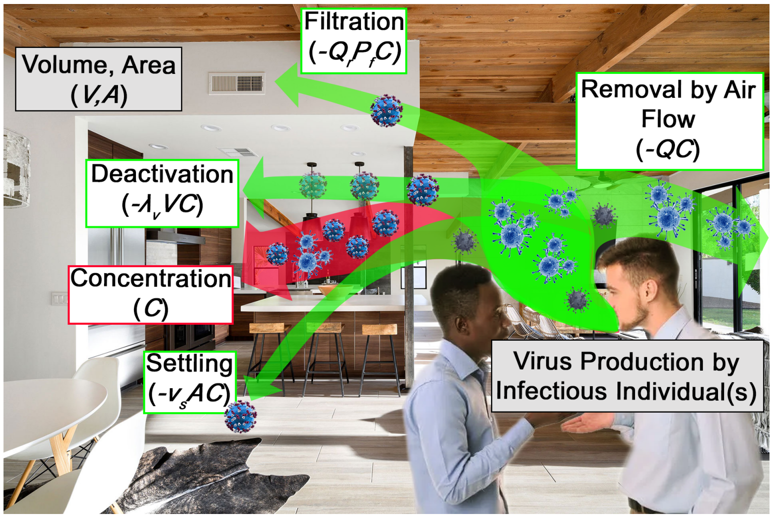


Figure 5: Shows how covid particles behave in a room with virus production from an infected individual.

The solution to this ODE can be found analytically, as it is a separable ODE. For simplicity, we first collect the coefficients into single parameters like seen in Eqs. (3, 4):

$$C_s = \frac{IP}{\lambda_c V} \quad \textbf{and} \quad \lambda_c = (Q + p_f Q_r + v_s A + \lambda_v V)/V \qquad (3, 4)$$

The ODE in Eq. (2) can then be rewritten as seen in Eq. (5):

$$\frac{dC}{dt} = C_s\lambda_c - \lambda_c \quad \Rightarrow \quad C\frac{dC}{C_s\lambda_c - \lambda_c C} = dt \tag{5}$$

This form of Eq. (5) can be integrated and simplified to find the closed-form Eq. (6):

$$\frac{-1}{\lambda_c} ln|C_s\lambda_c - \lambda_c C| = t + c \quad \Rightarrow \quad C_s\lambda_c - \lambda_c C = Ae^{-\lambda_c t} \quad \Rightarrow \quad C = C_s - Ae^{-\lambda_c t} \tag{6}$$

When assuming the infected person(s) enters at $t = 0$, the initial concentration $C(0) = 0$, which gives $A = C_s$, resulting in the following analytical solution in Eq. (7) [2]:

$$C(t) = C_s(1 - e^{-\lambda_c t}) \tag{7}$$

This has a similar form to the Wells-Riley model, but with explicit calculations for the concentration (analogous to quanta generation rate). This has a steady-state solution C_s, which can be used as a conservative estimate when the time since an infected person entered is unknown. The constant λ_c is the combined losses from all the sources of removal, and has units of 1/[time]; it can therefore be thought as the inverse of the time constant. Intuitively, this means that as the virus is removed faster (e.g. through higher air flow or more filtration), λ_c will increase and the time constant will decrease, so steady state will be reached faster. We can relate this concentration to risk by first calculating the dose d as shown in the following integral in Eq. (8) [2, 17]:

$$d(t) = \int_0^{\tau} Q_b C(t) p_m dt \tag{8}$$

Where p_m describes the probability of particles being removed by a mask. This can be related to the Wells-Riley model by using a parameter that relates the amount of pathogens to quanta, although this is not used in the model. In fact, in the case where $\lambda_v = v_s = Q_r = 0$ (meaning there is no viral deactivation, settling, or recirculation/filtration), this model simplifies exactly to the Wells-Riley model [3]. This can be seen by assuming an exponential dose-response, where the dose can be related to the risk of infection using a factor k (a rescaling factor that is based on the number of pathogens needed to cause an infection in an individual) as shown below in Eq. (9) [17]:

$$P_I = \frac{N_I}{S} = 1 - exp\left(\frac{-d(t)}{k}\right) \tag{9}$$

Of the existing tools for predicting COVID-19 risk, most rely either on statistics [9] or are based on this dose-response model [7, 8]. In particular, one of the major tools that we sought to improve upon in this project was a web app based on the paper this specific wording of the model is based on [7]. There is a similar spreadsheet implementation of the model that is also of interest to us. Although these tools provide strong guidance for implementing the model in practice, they rely on the well-mixed assumption which is stated as not always valid. Additionally, the models do not incorporate confidence intervals, which means the strength of the estimate is not obvious to the end user. The spreadsheet model also explicitly states in its FAQs that "people without a science or quantitative background have … trouble" using the tool, which indicates that a more accessible tool may be necessary to satisfy a wider audience [8].

Beyond Well-Mixed: The dose-response model is much more versatile than the Wells-Riley model, but it is still based on the well-mixed assumption (as concentration is not computed as a function of location). One major difficulty in moving beyond the well-mixed assumption, however, is that we could not construct an exact spatial distribution of the concentration, since the location of the infectious person(s) is not known. Any spatial distribution is based on the location of infectious people. Therefore, a probability distribution is used to model the spatial non-uniformity instead, which can provide upper and lower bounds to the concentration. The form of this distribution, according to Tan et al., is lognormal, meaning the natural log of the random variable X has a normal distribution with mean μ and standard deviation σ. The distribution of X can then be described with these three equations, Eqs. (10, 11, 12) [19]:

$$P(X) = \frac{1}{X\sigma\sqrt{2\pi}} e^{-\frac{1}{2\sigma^2}(ln(X)-\mu)^2} \quad \textbf{and} \quad \overline{X} = e^{\frac{\mu+\sigma^2}{2}} \quad \textbf{and} \quad X_{std} = \sqrt{\left(e^{\sigma^2} - 1\right) e^{2\mu+\sigma^2}} \quad (10, 11, 12)$$

where μ was found to be 0 and σ to be 0.9 [19]. This distribution can then be shifted and scaled to fit the concentration using the mean and standard deviation of the concentration. This scaling is described with the following Eq. (13):

$$C = \frac{X-\overline{X}}{X_{std}} C_{std} + \overline{C} \tag{13}$$

where $\overline{C}$ is the concentration from the dose response model and C_{std} can be found using the linear regression seen in Eq. (14) [19]:

$$\frac{C_{std}}{\overline{C}} = \beta_0 + \beta_1 \frac{\lambda_a}{\lambda_{a0}} + \beta_2 \frac{\sqrt{A}}{H} + \beta_3 \frac{H}{H_0} \tag{14}$$

By solving Eq. (15) for X in terms of C, Eq. (13) can be obtained:

$$X = \frac{C-\overline{C}}{C_{std}} X_{std} + \overline{X} \tag{15}$$

which can be combined with Eq. (10) in order to obtain the probability distribution of C. This allows us to find 95% confidence intervals of C, giving us the ability to take into account the spatial distribution of the concentration. By using these confidence intervals and Eq. (8) to find the upper and lower bounds of the dose d, the upper and lower bounds of the probability can be found through Eq. (9). The work defining these parameters and establishing the probability model are the largest contributions by the sponsor into this field. Properly implementing these equations is a major area of improvement for the app in the future.

Programming with Swift: Our initial prototype was completed for iOS, as all group members use iPhones so this enabled easier testing. The preferred language for iOS development is Apple's programming language, Swift, which is best utilized in the macOS exclusive IDE Xcode [2]. This IDE provides drag and drop tools to quickly develop a UI without needing to program each element, which will prove very useful for developing prototypes [13]. The language itself is an object-oriented language similar to Objective-C, so our training in C++ from ENGR 101 helped us with app development.

Information Sources: No engagement with the librarian was needed because Professor Capecelatro had knowledge in the field and provided many resources on the subject. In addition to peer-reviewed scientific and medical articles about infectious disease risk models [3, 15, 18, 19, 21] and the tools built on these models [7, 8], we have also collected many articles describing the parameters used in those models [2, 4, 5, 10, 11, 12, 16]. These sources were essential for getting numerically accurate results once work began on implementing the models. Additionally, interviews with Professor Capecelatro helped inform the motivation and goals for the project [6]. Other stakeholder interviews and surveys were conducted throughout the design phase as discussed in the stakeholder engagement section on p. 11. Finally, information about app programming has been researched in order to familiarize ourselves with app development and coding.

REQUIREMENTS AND ENGINEERING SPECIFICATIONS

Stakeholder inclusivity was used to assess relevant requirements. Relevant research was used to determine appropriate engineering specifications and methods for validation.

Requirements

Requirements were determined by interviewing the project sponsor to define the problem as concisely as possible. Table 1 on p.15 shows our prioritized list of requirements, specifications, and validation methods.

Table 1: Prioritized list of user requirements, engineering specifications, and the methods for validating those specifications.

Prioritized User Requirements	Engineering Specifications	Method for Validating Specification
1. Risk Assessment Accurately Modeled	1A. When using inputs from documented outbreaks, the actual infection rate lies within the confidence intervals for infection risk > 95% of the time	Validated by comparing model output with Excel risk-assessment model [8]
2. Easy to Use	2A. > 90% of users take < 60s to complete inputs 2B. Users enter app inputs accurately > 90% of the time	Validated by user testing input time and comparing user inputs to known room parameters
3. Easy to Interpret	3A. > 95% results are correctly interpreted by users	Validated by user surveys assessing their interpretation of the results
4. Transparent Calculations	4A. > 90% of users can find FAQs and relevant background information 4B. > 90% of users do not have further questions	Validated by user surveys assessing information accessibility
5. Quick Runtime	5A. Runtime for app computation is < 1s	Validated by the computational time of the code and time for software to generate risk probability

The most important requirement in our process was that the model is accurate. This is because the app would be useless if accuracy was not a factor, and it could potentially cause harm to users if false information was provided. The second most important requirement was that the app is easy to use. In order to provide maximum benefit to our users, it must be feasible to operate the app without any unknowable or overly complex inputs. The third requirement was the results must be easy to interpret. The results must be presented in a way that is easily understood by a majority of our users, otherwise it will have a smaller impact than anticipated. The fourth requirement, results should be transparent, is relevant to users that want to know more about how the results have been calculated, confidence intervals, and model parameters, and important to building trust with users. The last requirement was the app should have a quick runtime. This is important so that the app is convenient to use. In addition, users would be able to run multiple trials with varying parameters in a short period of time if they wanted to get a sense of their risk in different environments, or see what factors have the largest effect on changing risk.

The first three requirements listed (accurate, easy to use, easy to interpret) were necessary for the app to be successful. The final two (transparent calculations, fast runtime) were “wants” and not necessary, but are instead characteristics that improved the quality of the app.

Specifications

Each of the engineering specifications in Table 1 on p. 15 was determined based on maximizing app usefulness and pre-existing standards. For example, the first engineering specification says that the model accuracy needs to fall within 95% confidence intervals. This was chosen based on the existing standard in engineering and science of using 95%. The other four specifications were chosen to provide the maximum usefulness to our users.

In addition, the engineering specifications were reasonable. The first specification is backed by standard practice in the field and the next three specifications are mainly focused on user inputs and model parameters. These can be optimized with an intuitive user interface and organizational tools like drop down menus and sliders. The final specification deals with runtime and can be adjusted through code optimization.

Methods for Verifying Specifications

As seen in Table 1 on p. 15, the method for verifying the first specification was done through comparing the output of our model with the output from the Jimenez and Peng Excel model [8]. The next three specifications underwent verification testing using stakeholder tests to effectively analyze how users are interacting with the app. The final specification was verified by assessing the computational time it takes to generate risk probability.

CONCEPT EXPLORATION

The importance of concept exploration was to help us identify as many solution concepts as possible to spur innovation. The greater the diversity of ideas, the more likely we were to come across innovative ideas. In order to develop the best possible solution concepts, we first focused on divergent thinking. Divergent thinking is associated with creativity and stresses the importance of multiple solutions to a problem. Several divergent thinking techniques and concept generation methods can be used to more effectively identify solution concepts. These techniques and methods include brainstorming, mind mapping, design heuristics, morphological analysis, functional decomposition, and TRIZ. Because of the nature of the project, morphological analysis and mind mapping were easiest to utilize.

Concept Generation Methods

Multiple divergent thinking techniques were used throughout the concept exploration stage, specifically for concept generation. Some of these techniques include brainstorming, mind mapping, and morphological matrices.

Brainstorming: Initially we began exploring the ideation space by brainstorming individually until we had reached forty ideas each. This was helpful because it prevented us from distracting each other and allowed for a large quantity of ideas produced at once. However, this quickly became disorganized with no clear distinction between ideas for the different subsystems of our project. As a result, when we came together after this initial brainstorming session, we focused on comparing and organizing our ideas.

Mind Map: A mind map was used to clearly show the organization of our ideas for user input questions. These are vital to how users will be interacting with the app, making them a critical subsystem. If our app asked questions that the users do not know the answers to, they would not be able to effectively assess their risk. As a result, it was imperative that we were able to correctly identify concepts that allow for users to interact with the app in an intuitive manner. The mind map created can be seen below in Fig. 6. Mind mapping for user input questions was able to produce over 30 distinct ideas and organize these ideas according to which model parameter they relate to.

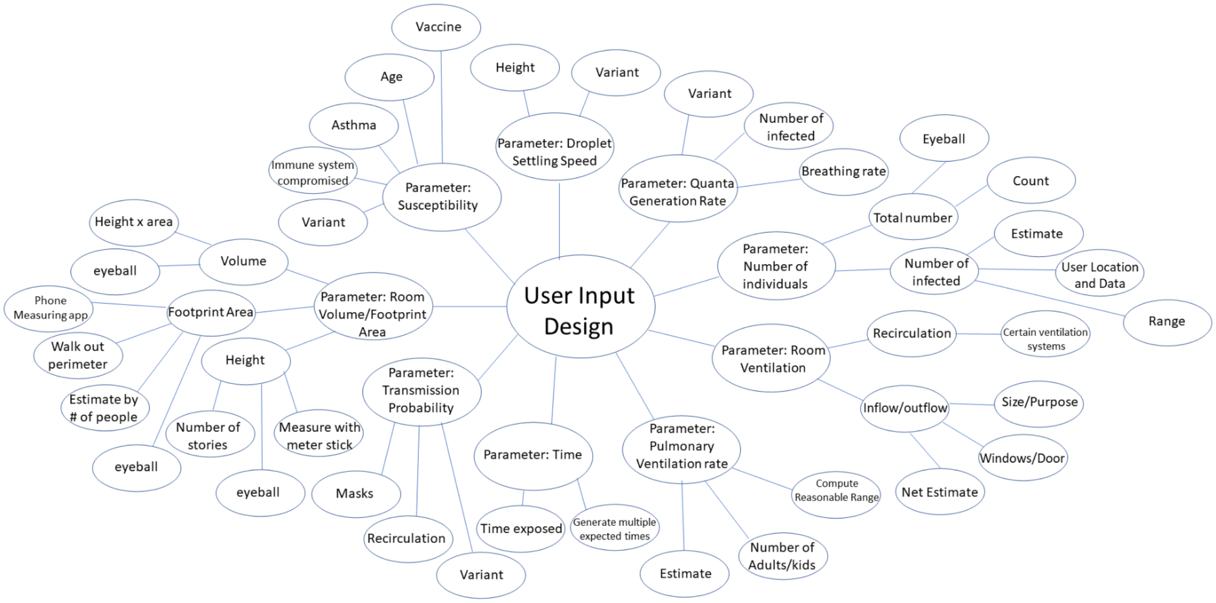


Figure 6: Shows the mind map created to organize concepts generated for ideas related to user input questions.

Morphological Analysis: Morphological analysis is another tool that was used to generate concepts primarily for the user interface. Fig. 7 below shows a simple morphological matrix that provides ideas for how users will enter information into the app.

Subsystem	Idea 1	Idea 2	Idea 3
Input method (a)	Text boxes	Sliders	Discrete Categories
Input type (b)	Directly using parameters	More intuitive Abstractions	Room Type
Input Error Estimation (c)	Slider	Qualitative Questions	

Figure 7: Shows an example morphological matrix for the user interface for inputs.

For example, for a simple parameter such as room volume, the user could enter in the number of cubic feet into a text box, there could a slider present so that the user can quickly approximate the volume, or there could be discrete categories for room volume like small, medium, and large that correspond to predetermined values. There would likely be more associated uncertainty with discrete categories; however, the benefit would be simpler user inputs.

Concept Generation Results

As demonstrated previously in Figs. 6 and 7 and again in Appendix A, these concept generation methods yielded many diverse ideas. Using the example of room volume again, we thought of incorporating the measuring tool that is present by default in iPhones so users would be able to take a picture of the room and have the measuring tool find the room dimensions. We also thought of posting QR codes on certain rooms with known parameters to allow for a user to scan the code and import all necessary information including room volume and ventilation rate. Further ideas that help to demonstrate concept diversity involve identifying the number of infected individuals. For example, we had the idea to include statistics of local infectivity rates to estimate the number of infected people or estimate based on the square footage of the room.

After analyzing these ideas, we determined that there were three main subsystems: concepts for user input questions, concepts for the user interface (UI) for inputs, and concepts for the presentation of output results. For the user input questions, we essentially determined two categories of questions; quantitative questions that directly asked the user to input the actual parameter value, and qualitative questions that asked the user some more intuitive question about the room that would then be correlated to a specific parameter.

The most important UI design concepts can be seen below in Fig. 8. We considered various user interface design choices such as text boxes, sliders, drag and drop options, drop down menus, preset options, and discrete categories to help users interact with the app in the most intuitive way possible.

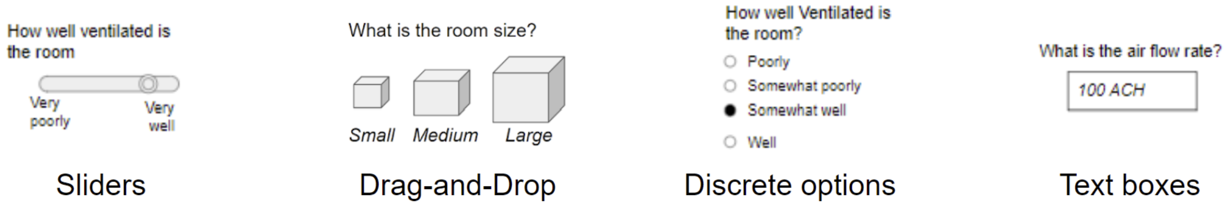


Figure 8: Shows different user interface options for inputs.

Finally, for the result output, we did not arrive at discrete ideas like the concepts for UI or input questions, but instead had a sliding scale of detail provided to the user. The project requires that we show the numerical risk assessment with confidence intervals. We developed additional concepts such as a qualitative risk assessment that indicates if a given risk level is high or low, color coding the presentation red for high and green for low, and including a comparison to a familiar activity to make the level of risk clearer.

Concept Selection Process

The first factor used to narrow the list of generated concepts was feasibility. The main considerations for feasibility were project deadlines, computation time, and coding complexity. For example, an idea for a real-time 3D model of the room as the user enters inputs was determined not to be feasible based upon the computation time and our coding knowledge. A user survey was used to further narrow the concept list of user questions through empirical testing to aid in evidence-based concept selection. This survey has shown that questions that quantitatively ask for the direct model parameters will likely not be accessible, and questions that qualitatively ask about the physical parameters of the room will be much more useful.

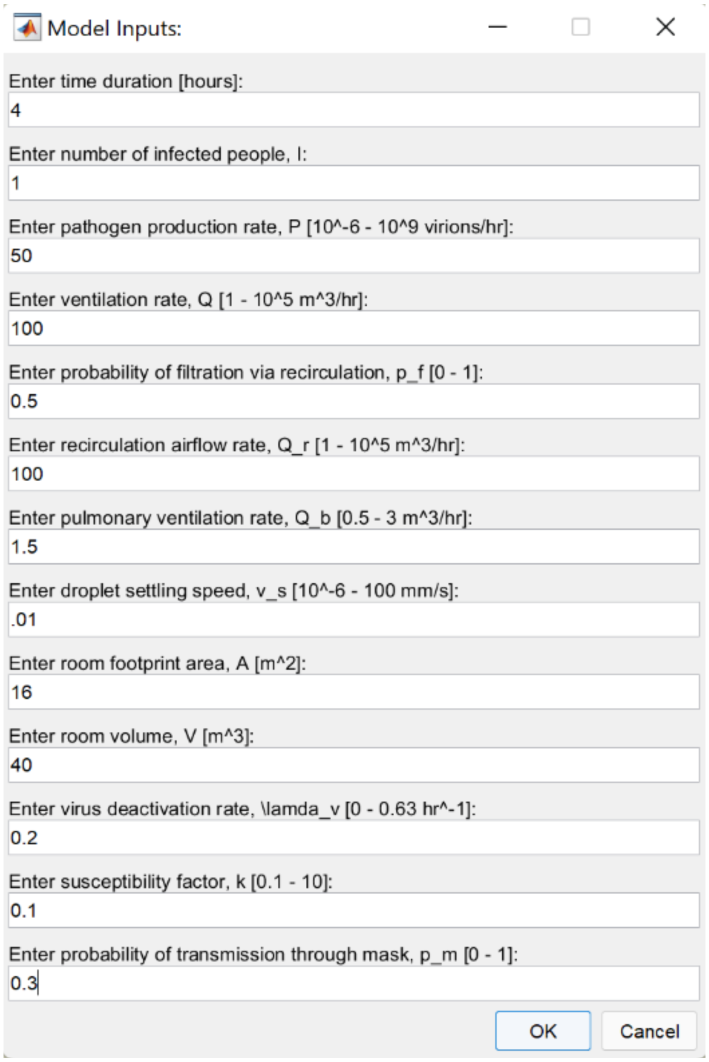
After surveys for input questions, we performed user tests where users enter inputs for a room with known parameters on Tuesday, April 12th. For input UI, speed of entering results was used to determine the best concepts. This testing was done using the project advisor, Professor Kaviany's ME 335 class in room 2505 GGB.

As demonstrated, much of our concept selection process was determined by stakeholder surveys and tests to optimize how users interact with the app. This allowed for iteration and additional concept generation as necessary. All selected concepts were consistent with engineering specifications and user requirements. Fixation was not a problem because the ideation space has been widely explored and stakeholder testers have a significant degree of power when it comes to these user interface design choices. We implemented what our stakeholder tests indicated were the optimal solutions.

Alpha Design

A simple alpha prototype of the app has been iterated multiple times, and a functional alpha prototype has been established in MATLAB using the dose-response model from Eqs. (7-9). This type of simple alpha design to test model behavior was recommended and heavily influenced by the project sponsor. Initial prototypes were developed in Google Sheets and Microsoft Excel, but the lack of base integral functions in these softwares led to performance issues. Therefore, the alpha was developed in MATLAB, and has been iterated in MATLAB to further improve performance. The main purpose of the alpha design was to verify model behavior and to have a programmed version of the model that could be translated to Swift for a beta design that will be used in user testing.

The alpha design is a purely virtual embodiment of a prototype solution for computational analysis, paralleling the virtual embodiment of the final solution which is a smartphone app. For simplicity and to develop an alpha design prior to finishing user input selection, this prototype uses direct model inputs (e.g. ventilation rate, Q [m^3/hr], or the room volume, V [m^3]) rather than abstract and qualitative inputs. The pop-up box used in the alpha design to allow users to directly input the model parameters can be seen with arbitrary input values in Fig. 9 p. 20.



Enter time duration [hours]:
4

Enter number of infected people, I:
1

Enter pathogen production rate, P [10^-6 - 10^9 virions/hr]:
50

Enter ventilation rate, Q [1 - 10^5 m^3/hr]:
100

Enter probability of filtration via recirculation, p_f [0 - 1]:
0.5

Enter recirculation airflow rate, Q_r [1 - 10^5 m^3/hr]:
100

Enter pulmonary ventilation rate, Q_b [0.5 - 3 m^3/hr]:
1.5

Enter droplet settling speed, v_s [10^-6 - 100 mm/s]:
.01

Enter room footprint area, A [m^2]:
16

Enter room volume, V [m^3]:
40

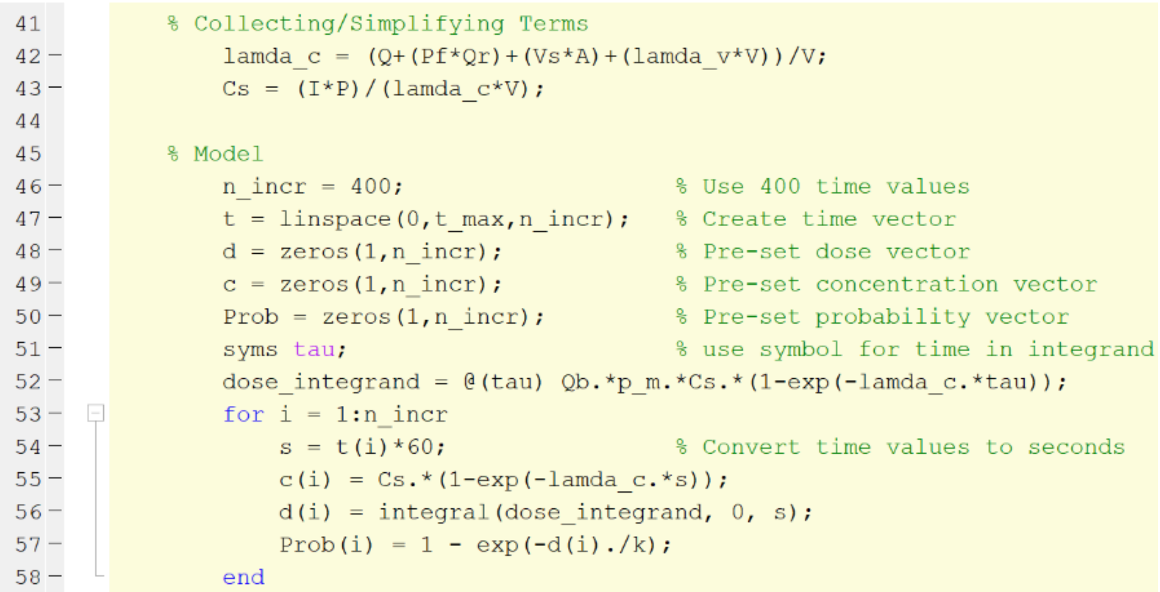
Enter virus deactivation rate, \lamda_v [0 - 0.63 hr^-1]:
0.2

Enter susceptibility factor, k [0.1 - 10]:
0.1

Enter probability of transmission through mask, p_m [0 - 1]:
0.3

Figure 9: Pop-up box from alpha design, allowing users to input all necessary model parameters. Note that the input values used are for an arbitrary room.

The alpha design iteratively computes the concentration using Eq. (7), the dose using Eq. (8), and the probability of infection using Eq. (9). A snippet of the alpha design MATLAB code that includes the for-loop that computes the concentration, dose, and probability of infection can be seen in Fig. 10 on p. 21.



```
41       % Collecting/Simplifying Terms
42 -         lamda_c = (Q+(Pf*Qr)+(Vs*A)+(lamda_v*V))/V;
43 -         Cs = (I*P)/(lamda_c*V);
44
45       % Model
46 -         n_incr = 400;                       % Use 400 time values
47 -         t = linspace(0,t_max,n_incr);       % Create time vector
48 -         d = zeros(1,n_incr);                % Pre-set dose vector
49 -         c = zeros(1,n_incr);                % Pre-set concentration vector
50 -         Prob = zeros(1,n_incr);             % Pre-set probability vector
51 -         syms tau;                           % use symbol for time in integrand
52 -         dose_integrand = @(tau) Qb.*p_m.*Cs.*(1-exp(-lamda_c.*tau));
53 -         for i = 1:n_incr
54 -             s = t(i)*60;                    % Convert time values to seconds
55 -             c(i) = Cs.*(1-exp(-lamda_c.*s));
56 -             d(i) = integral(dose_integrand, 0, s);
57 -             Prob(i) = 1 - exp(-d(i)./k);
58 -         end
```

Figure 10: Critical portion of the alpha design MATLAB code, where concentration, dose, and probability of infection are computed iteratively in lines 55-57, respectively.

These calculations for concentration, dose, and probability of infection are done for 400 linearly spaced times between zero seconds after the room is occupied by an infected person(s) and the time duration that the user inputs. The engineering analysis for this model at this stage consisted of analyzing the qualitative behavior of the graphs and confirming that it meets what is expected of the equations. The behavior of the concentration as computed by the alpha design for an arbitrary room can be seen in Fig. 11 below.

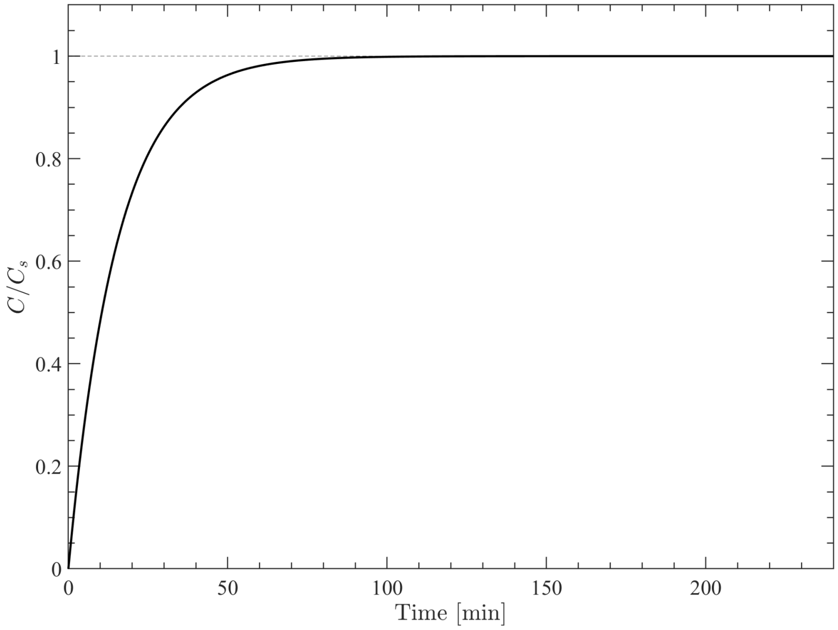


Figure 11: Concentration, normalized by the steady-state concentration, as a function of time. The normalized concentration function meets expected behavior of an exponential function that approaches unity at steady state.

Through simple analysis of the governing first order ordinary differential equation for concentration as seen in Eq. (2) and the solution to this differential equation as seen in Eq. (7), it can be seen that the behavior of the normalized concentration meets expectations. An exponential function that approaches unity at steady state is both expected and seen in the normalized concentration function behavior as produced by the alpha model. Additionally, using these parameters, $\lambda_c = 0.0011\ s^{-1}$, and therefore the time constant for concentration is approximately 15 minutes. Thus, concentration is expected to be at 98% of steady state at the time of four times the time constant, 60 minutes. This expected behavior is seen in the computations done by the alpha design. The dose function as computed in the alpha design also meets expectations, as seen in Fig. 12 below.

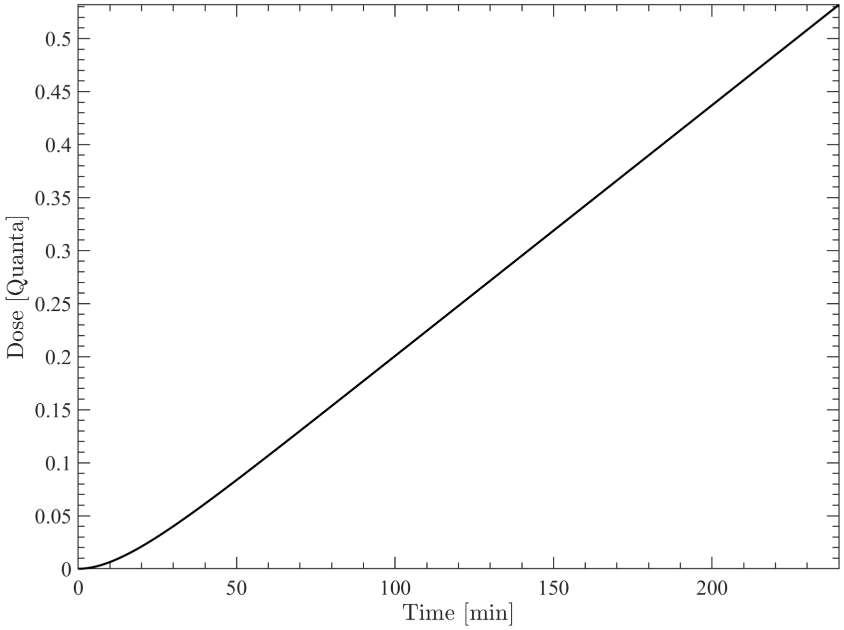


Figure 12: Dose as a function of time as computed by the alpha design meets expectations, as a steady-state behavior is met simultaneously to the concentration steady-state.

The dose function as computed by the alpha design meets expectations and approaches a linear steady-state behavior, starting around 60 minutes. This is the expected qualitative behavior for dose, as concentration is approximately at steady-state by 60 minutes with these model parameters, and therefore the integral for dose is expected to behave approximately linearly at this point through simple analysis of this integral. The probability of infection as a function time as computed by the alpha design can be seen in Fig. 13 on p. 23.

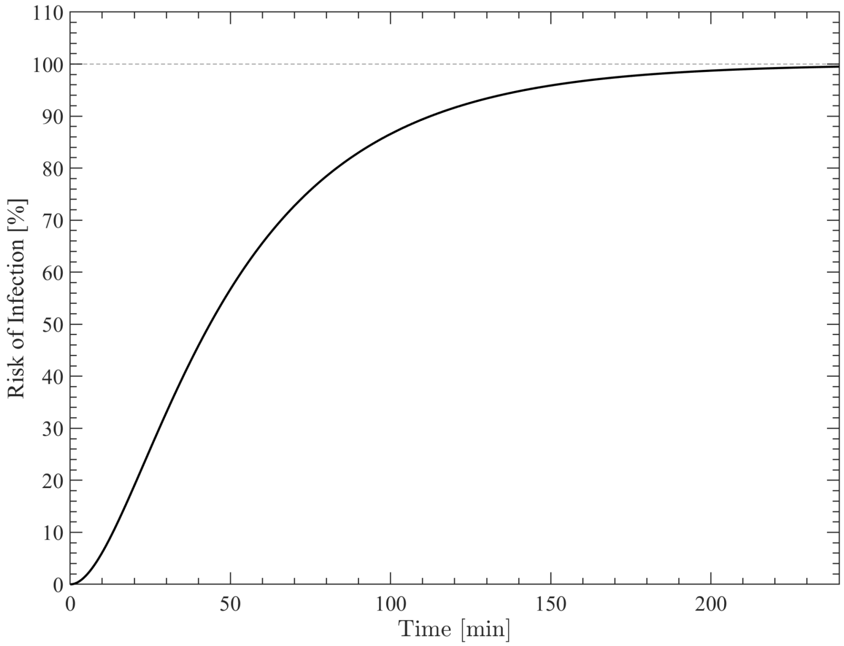


Figure 13: Risk of infection [%] as a function of time, as computed by the alpha design. Sigmoid behavior is seen, and risk of infection approaches a steady-state value of 100% as time goes on, which meets the expected qualitative behavior.

The risk of infection, as computed by the alpha design and presented as a percentage, meets expectations and approaches a 100% risk at steady-state. Sigmoid behavior is seen and expected due to the dose integral being in the exponential in Eq. (9) for the probability of infection.

Validation: As previously mentioned, the purpose of this alpha design was primarily to verify proper programming of the model, verify function behaviors, and do initial testing for requirement and engineering specification 5 (computation time is less than 1 second). The programming and function behaviors have been verified above, and initial testing for the alpha design computation time has been performed. The alpha design took approximately 5 seconds for computation, and therefore did not meet the specification of a runtime less than 1 second. However, the initial iterations of the alpha, both in Google Sheets/Microsoft Excel and MATLAB, had computation times of approximately 30 seconds. The runtime was reduced after meeting with the project sponsor and more efficient coding methods were discovered. Also, the alpha design performed iterative calculations for 400 distinct time points for debugging and demonstration purposes, but further prototypes and the final design only calculated once for the time value input from the user. Therefore, the beta and final design easily met the specification of a computation time of less than one second.

It is important to note that the alpha design was unable to quantitatively test the prototype for requirements and engineering specifications 2-4 (Easy to use, Easy to interpret, and Transparent calculations), although requirement 1 had been tested qualitatively and requirement 5 had been been tested empirically, as discussed above. Because the main purpose of this alpha design was to qualitatively validate the mathematical models, it was not useful in helping to select from the concepts discussed in concept generation (i.e. UI elements or input questions). In fact, as a MATLAB model, it was unable to incorporate most of the app UI concepts, and the input question concepts did not have time to be implemented because concept selection through user surveys was ongoing until Thursday, March 16th. Instead, the UI concepts and input questions were selected for a beta release of the model, built in the form of an app, which has been completed and the model was numerically verified.

Concept Analysis and Iteration

Concepts were selected and finalized, and therefore a thorough engineering analysis of concepts with respect to their ability to meet requirements and specifications was conducted. However, some concepts were ruled out due to lack of feasibility. Additionally, user survey data showed that the user input concepts should be qualitative, and therefore user interface concepts should be cohesive with qualitative user inputs. The user survey results can be found in reference [20].

These concepts were verified for meeting most engineering specifications most quickly and easily through user testing and follow-up surveys. More specifically, the ability of concepts to meet engineering specifications for requirements 2-4 (Easy to use, Easy to interpret, and Transparent calculations) were easily tested for in this manner. User testing was conducted in room 2505 with Professor Kaviany's ME 335 class, which has known model parameters, and participants had the chance to use a version of the beta design app to empirically test if the concepts used in the beta version met these engineering specifications. This testing was done on Tuesday, April 12th.

The ability of concepts to meet engineering specifications for requirements 1 and 5 (Accuracy of model and a quick runtime) was most quickly and easily tested through our own internal testing. For the engineering specification for requirement 1, we compared the output of our model with the output from the Excel spreadsheet by Jimenez and Peng [8]. For the engineering specification for requirement 5, we used the selected concepts in a prototype and analyzed the computation times for a wide set of varying inputs, and ensured that the runtime remained below the threshold of one second.

ENGINEERING ANALYSIS

A detailed engineering analysis was conducted to determine the optimal combination of design choices. A combination of computational and empirical methods were used to analyze the options for model parameters, user inputs, and the user interface. For the model parameters, a computational sensitivity analysis was performed to get a better understanding of the importance of each parameter on the model behavior. Then the least sensitive parameters underwent additional testing to determine if the removal of these parameters resulted in a significant offset. For the user inputs, empirical testing was conducted via stakeholder surveys to identify how comfortable users would be with inputting various parameters. The user interface was analyzed empirically as well through additional user testing for design verification. For these tests, students in 2505 GGB were asked to enter the room parameters, which had already been identified. The survey results were analyzed by accuracy and time to completion.

Sensitivity Analysis

In order to understand the effects of changing the model inputs on the risk assessment output of the dose response model that is used, a computational sensitivity analysis has been conducted. In this analysis, sensitivity is defined as the ratio of the resulting percent change in risk due to a percent change in the given input parameter. This analysis was primarily used to determine the parameters for which the uncertainty must be minimized, and secondarily if any model reductions could be made. The parameter values that have been used as the baseline from which percent changes were measured from can be found in Appendix B. The results of the sensitivity analysis in order of greatest to least sensitivity can be found in Table 2 below.

Table 2: Sensitivity analysis conducted for all model parameters. The red highlighting shows which parameters have insignificant sensitivity, and because of their limited impact, model improvements have focused on ways to minimize these variables.

Model Parameter	Parameter Meaning	Sensitivity
k	Susceptibility Factor	0.9391
I	Number of Infected	0.8626
P	Pathogen Production Rate	0.8626
Q_b	Pulmonary Ventilation Rate	0.8626
p_m	Inhalation through Mask	0.8626
V	Room Volume	0.5716
Q	Ventilation Rate	0.2911
λ_v	Virus Deactivation Rate	0.0921
p_f	Filtration by Recirculation	0.0147
Q_r	Recirculation Airflow Rate	0.0147
v_s	Droplet Settling Speed	0.0016
A	Footprint Area	0.0016

From the sensitivity analysis, it is seen that many model parameters have the same sensitivity as another parameter. By evaluating Eqs. (2) and (4), this trend is expected for any parameters that are used in a product with another parameter. It is important to note that the importance of the uncertainty in parameters with higher sensitivities is significant. Therefore, in the finalization of model parameters, it was important to minimize uncertainty for these more sensitive parameters. We have found that p_f , Q_r, v_s, and A have negligible sensitivities, meaning that changing these parameters leads to less than a 5% change in the model output, so model reductions focused around these parameters have been considered.

Model Reductions

As previously mentioned, the sensitivity analysis has allowed for potential model reductions to be made. Specifically, neglecting the contribution of droplet settling speed, v_s, as well as combining P, the pathogen production rate, with k, the susceptibility factor, and using this new P as the quanta generation rate instead has been considered.

Neglecting the droplet settling speed was considered as a model reduction for multiple reasons. First, the droplet settling speed depends on particle size and has a large uncertainty due to many factors, so removing this term can aid in reducing uncertainties in the model. Second, the removal of droplet settling speed leads to a more conservative risk estimate, as settling speed in the model acts as a pathway for pathogen removal from the enclosure, so risk is increased by neglecting the droplet settling speed. By neglecting the droplet settling speed, Eq. (4) is reduced, and the reduced version can be seen in Eq. (16).

$$\lambda_c = (Q + p_f Q_r + \lambda_v V)/V \tag{16}$$

In order to ensure that removing settling speed would not significantly affect the model output, further analysis was conducted. Initial analysis results can be found in Table 3. In this analysis, the same default parameters from Appendix B were used, and v_s was varied. The full model is considered to be when Eq. (4) is used in the model, while the reduced model is considered to be when Eq. (16) is used. This analysis was conducted for times of 30 minutes, 60 minutes, and 120 minutes, then the percent differences were averaged.

Table 3: Comparison between the full model and the reduced model to verify that the reduced model is applicable for small settling speeds, or those associated with small particle sizes. Increase in risk associated with considering settling speed negligible using different values of v_s

Value for v_s [mm/s]	Time [min]	Full Model Output	Reduced Model Output	Average % Difference
0.01	30	0.1394	0.1396	0.174
	60	0.3547	0.3554	
	120	0.6645	0.6657	
0.10	30	0.1377	0.1396	1.717
	60	0.3486	0.3554	
	120	0.6538	0.6657	
1.00	30	0.1220	0.1396	17.77
	60	0.2961	0.3554	
	120	0.5601	0.6657	

From this table, it can be seen that the difference is only significant in this setup for settling speeds on the order of 1 mm/s. However, using the Equation for the Stoke's settling speed from the Bazant paper [7], and the peak droplet density at 1.5 um, we calculated that the maximum droplet settling speed for most of the COVID droplets will be 0.23 mm/s. This is much smaller than the 1 mm/s that caused significant error, so we conducted additional tests on whether or not excluding this maximum reasonable value would affect the results. When calculating the risk both with and without this settling speed, we found that excluding the droplet settling speed only caused a 10% increase in risk (when using a reasonable room geometry where the volume was at least 2.5 times greater than the area, representing a room height of at least 2.5m). Additionally, this error was only high when the situation was already a risky environment, such as a very small room with poor ventilation. These are the scenarios where we would want to be most conservative. Because excluding the droplet settling speed only results in large error in situations where we desire a conservative estimate, and because consolidating it into one variable carries a large amount of uncertainty, we consider excluding it to be a reasonable choice.

Another model reduction that has been made is neglecting k, the susceptibility factor, by using P as the quanta generation rate rather than the pathogen production rate. This is because k acts as the conversion factor between quanta generation rate and pathogen production rate, with units of pathogens/quanta. Although k does vary based on factors such as user age, immune status, and vaccination status, much of the literature uses it as a constant, and this constant is inconsistent within the literature. Therefore, eliminating the parameter removes some explicit uncertainty within the model. Additionally, there is much more literature published for quanta generation rates than pathogen production rates, so this reduction aids in parameter finalization for the model. This reduction is applied to Eq. (9), and yields Eq. (17):

$$P_I = \frac{N_I}{S} = 1 - exp(-\ d(t)) \qquad (17)$$

It is important to note that in Eq. (9), the dose, *d(t)*, is in the units of pathogens and converted to quanta by the susceptibility factor, while in Eq. (17) the dose is in units of quanta and does not need a conversion factor.

FINAL DESIGN DESCRIPTION

The current design is an iOS app prototype that asks for the parameters described in the model reduction section, and uses the equations from the MATLAB alpha. In order to obtain values for these parameters, we chose to ask qualitative questions when possible, as the stakeholder survey indicated that people would not be able to estimate most of the quantitative parameters (with the exception of room footprint area). The other quantitative questions we chose to ask were for obtaining the time spent in the room, as this parameter is completely up to the user, and the number of infected people in the room, as this is an unknown quantity that the user would be best able to estimate. Some parameters were combined into one question in order to simplify user input. Specifically, the question for P and Q_b were combined as both depend solely on activity level (although there is an option for a separate activity level if the user is doing a different activity to others in the room), and p_f, Q_r, and Q were all combined into a single question about ventilation type. This gives us 7 required questions with one optional question, which can be seen in Table 4 on p. 28.

Table 4: Questions app asks users to qualitatively determine input parameters

Model Parameter	Parameter Meaning	Question
I	Number of infected individuals	How many people do you think are COVID-19 positive in the room?
P	Quanta Generation Rate	What type of activity are people participating in?
Q_b	Pulmonary Ventilation Rate	What type of activity are you participating in? **(OPTIONAL)**
Q, p_f, Q_r	Ventilation rate, Probability of Filtration by Recirculation, Recirculation Airflow Rate	What type of ventilation does the room have?
p_m	Probability of Inhalation through Mask	What kind of mask are you wearing?
V	Room Volume	How large is the room?
A	Room Area	What is the square footage? (used for determining volume and in the probability density function)
t	Time spent in the room	How long do you plan to spend in the room?

The wording of these questions was chosen as we think it is the most straightforward way of asking for these parameters; however, they can easily be changed in response to user feedback. These questions have been implemented into the app in a form that asks one question per screen, in the order presented above. The actual UI input method for each question can be seen in Table 5 on p. 29.

Table 5: User Interface elements used for each parameter within the app

Model Parameter	Question Interface Structure
I	Text box
P	Discrete options (related to activity level)
Q_b	Discrete options (related to activity level)
p_m	Discrete options (related to mask type)
V	Slider (small to large)
Q, Q_r, p_f	Discrete options (related to airflow)
A	Text Box
t	Text Box

These methods were chosen either as a natural extension of the question type. For example, quantitative inputs require a text box to input, while a continuous qualitative estimate like volume works well with a slider.

The app starts on an intro screen that gives some background information on the app and a link to an info screen. The info screen contains more detailed background on the app as well as a link to a condensed version of this report. The intro screen also contains a button to start the app, where the user goes through the questions described above. Finally, there is a calculator screen where the user receives the estimated risk with confidence intervals, as well as another link to the info screen and an option to reset. Representative screenshots of the app can be seen in Figure 14 below, with screenshots of all screens in Appendix C


How many people do you think are COVID positive in this room?

What type of activity are other people participating in?

Are you doing the same activity?

How large is the room?

Your Risk of COVID-19 is:
0.021%
With Error bounds of
0.017% - 0.028%

Want to know how this risk was calculated? Press the "More Info" button below for some background information

Figure 14: Screenshots of the app showing the questions for I, P, V, and the final results screen.

The app has been verified against the MATLAB model to within 3 significant figures, and we believe the code has been implemented properly. The biggest challenges with implementing the model were obtaining the exponential function and approximating the integral for dose, as neither were default functions in Swift. For the exponential function, we were able to find a library (Swift Numerics) that contained the exponential function, while the integral was approximated using a Left Riemann Sum with 400 evenly spaced steps. The accuracy and speed of these methods indicate these are reasonable implementations. The confidence intervals were added by applying uncertainty to the user inputs. For the discrete categories, the error bounds were generally 50% of the value or half the distance to the next value, whichever was smaller. For the continuous variables, the error was generally 25%. A full table showing the error bounds for each input can be seen in Appendix D.

Other challenges implementing the app included ensuring the transitions between screens passed the data and used the correct class types, and ensuring that connections between the UI and code were consistent. The data was passed between screens by setting up a separate class file that contained all the parameters, and including a variable of that class type within each screen. This variable can then be assigned to the next ViewController (the code file for each screen) using the `prepare(for: segue)` function in Swift. The connections between UI elements and code caused issues because when copying and pasting similar screens for ease of design, the UI elements continued to reference the code from the previous screen. Once this issue was discovered, deleting the erroneous connections solved the problem. To make the UI look more professional, a teal background was added and a Block M was put in the lower left corner. In order to make more elements fit on smaller screens, this was moved around and resized on different screens. Making the size and location uniform is an area of improvement to the UI. Additionally, we chose to round the corners on the buttons and changed the font to DIN Alternate Bold in order to give a more stylized appearance to the app. A more detailed description of the app structure and issues faced in making the app can be seen in the README in Appendix E.

Improvements to the app will be primarily focused on making it more generally useful, both from a programming perspective and from a modeling perspective. From the programming perspective, more work needs to be done on the UI to ensure it works properly across all Apple devices, which is a requirement for getting the app approved for distribution on the App Store. Some work has been done on this by using AutoLayout options for spacing text and input methods, but making the app work on extremely small screens will likely require making larger changes like adding scroll views. The app has also not been checked for accessibility guidelines like readability, so ensuring it meets those requirements is another possible design improvement. Additionally, making an Android version of the app would also greatly increase accessibility. From a modeling perspective, the app is currently hard coded with text and quanta generation rates for COVID, but the models used can be applied to any airborne disease. Making the text and parameters work for multiple diseases will also help make the app more applicable. Additionally, the current implementation of confidence intervals is generated using uncertainty for the inputs and not the probability density function described in the introduction. Although this was recommended by the project sponsor, a more sophisticated implementation of confidence intervals would help improve the app. Adding an option to the final calculation screen where users can edit specific parameters to see how that changes the risk would also be a helpful addition. Finally, adding some of the design ideas to the presentation of the risk, such as color coding the results or adding comparisons, will help with interpretation of the results. This will require setting thresholds for high, low, and medium risk.

The next steps for the app are primarily focused on the UI and implementing the beyond well-mixed model. First, we need to add the probability density function from the Beyond well-mixed model to the MATLAB model to verify we have implemented it correctly, and then translate that into Swift. Once the MATLAB model is fully verified in Swift, the debug information will be hidden and the layout of each UI element can be positioned properly. Ensuring the layout stays consistent across multiple types of iPhone and that the font size remains large and easily read will likely be the largest challenge. Additionally, we plan on adding some design ideas for the presentation of the results, like color coding the percentage red for high risk and yellow for medium risk. We will also be implementing more branding into the app to create a more professional appearance, including the use of a block M in the corner and more refined buttons and text. Previous implementations of the block M caused compiler errors, so figuring out how to properly build the app with external images will be important after the model is complete. To meet these goals, we will assign additional team members to the app development portion of the project as verification and validation steps get completed.

FINALIZED PARAMETERS RANGES AND VALUES

In order to allow for the qualitative questions in the current design to represent quantitative values for model parameters, the model parameters have a set number of realistic options that we have determined for values that can be used within the model. These finalized model parameter values and ranges can be seen in Table 6 on p.32.

Table 6: Finalized parameter values

Model Parameter	Parameter Meaning	Value(s)/Range	Units
I	Number of Infected	Direct user input	count
P	Quanta Generation Rate	Resting, oral breathing: **3.1** Heavy activity, oral breathing: **21** Light activity, speaking: **42** Light activity, singing (or speaking loudly): **270**	quanta/hr
Q_b	Pulmonary Ventilation Rate	Standing (office, classroom): **0.54** Talking (meeting room, restaurant): **1.1** Light Exertion (shopping): **1.38** Heavy Exertion (sports): **3.3**	m^3/hr
p_m	Probability of Inhalation through Mask	Cloth: **0.5** Medical Surgical: **0.10** N95: **0.05**	N/A
V	Room Volume	Direct user input	m^3
Q	Ventilation Rate	Closed windows: **0.3** Open windows: **2.0** Mechanical ventilation: **3-8** Open windows with fan: **6** Restaurant/bar: **9-15** Hospital/subway car: **18** Airplane: **24**	ACH
λ_v	Virus Deactivation Rate	**0.63**	hr^{-1}
p_f	Probability of Filtration by Recirculation	Residential Window AC - MERV 2: **0.2** Residential/Commercial/Industrial - MERV 6: **0.5** Residential/Commercial/Hospital - MERV 10: **0.85** Hospital - MERV 14: **0.90** HEPA - MERV 16: **0.95**	N/A
Q_r	Recirculation Airflow Rate	Slow: **0.3** Moderate (most small buildings/rooms): **1.0** Fast: **10** Airplane: **24** Subway car: **54**	ACH

As seen in Table 6, most of the finalized parameters are categorized in order to best suit the qualitative questions used in the current design. These qualitative questions were selected through stakeholder surveys in the concept selection process.

DESCRIPTION OF VERIFICATION AND VALIDATION APPROACH

Plans to verify our design against our requirements and engineering specifications, as well as to validate our design against the overall project problem, have been considered and developed. It is important that the design meets all of the original requirements and specifications, and therefore verification plans have been made to test the design for all specifications. Validation has begun, and will require additional work in the form of feedback from users through the published design.

Verification Testing Results and Future Plans

Verification tests have been planned to assess all of our requirements and specifications. Several have already been conducted, however, due to time and resource constraints, we were unable to finish verifying all requirements and specifications. The first verification test conducted was to determine whether our model is accurate with 95% confidence intervals. In order to accomplish this, we compared our model to the excel spreadsheet developed by Jimenez and Peng [8]. Various test cases were conducted with a wide range of inputs to ensure a complete model comparison. Once we had the outputs of the two models utilizing the same inputs, we were able to determine the percent error. Appendix F shows a table summarizing these verification results. When we added the confidence intervals to our model it was easy to ascertain that the already verified excel model lied within our confidence intervals 95% of the time. Since our model is accurate within 95% confidence intervals, our model was verified and the requirement and specification for accuracy was met. Verification by comparison to already verified models was the most convenient way to test the accuracy of our model. Although not the most rigorous analysis, it still provided valuable information and was more repeatable than comparing our model to known outbreak data. We were confident that we correctly used the work by Peng and Jimenz to assess the accuracy of the outcomes of our model. It is important to note that the outlier in the verification testing, test number 5 highlighted in red in Appendix F, was one of several extreme edge cases meant to test the limits of our model. It is likely that one of the assumptions we made led to a noticeable offset when parameters were taken to extreme circumstances. An alternative plan that could be conducted given more time is to compare the output of our model with known outbreak data.

The verification test for requirement 2, the app is easy to use, was assessed using stakeholder tests. On Tuesday, April 12 from 10:00-10:20 AM we utilized the students from Professor Kaviany's ME 335 class in 2505 GGB. The students were asked to use the beta version of the app to assess their risk for COVID-19 infection. They were asked to input room parameters including ventilation type, room size, square footage, activity performed in the room, and time spent in the room. By contacting University Facilities Services, we determined that 2505 GGB has an area of 3064 ft^2 or 284.66 m^2, a height of 14.67 ft or 4.47 m, and a ventilation rate between 13.3 - 19.7 ACH. With these known values and given more time, we would have been able to determine how accurately users enter this information. We would have determined the percent error between the values entered by users and the values provided to us by the University. We also analyzed the time it took for users to enter in the parameters. We determined that users were able to enter parameters in under 60 seconds, however, we were not able to determine whether these parameters were within 90% of the correct value. Therefore requirement and specification for 2A was met, but the requirement and specification for 2B requires additional analysis.

The verification test for requirement 3, the results are easy to interpret, was planned to be assessed using stakeholder tests as well. We planned to provide a link to the students in professor Kaviany's class that opens up a google form asking questions about the interpretation of results. If more than 95% of people were able to correctly interpret the results from the app then the interpretation will be verified and the requirement and specification would have been met. If 95% of people cannot correctly interpret results then the requirement and specification would not have been met.

The verification test for requirements 4, transparent calculations, was planned to be assessed using empirical stakeholder testing as well. We intended to send out a survey to the students that assesses their ability to understand how the calculations were performed. If more than 90% of users were able to navigate to the FAQ section of the app and had no further questions after reading the FAQs then this requirement and specification would be verified. If less than 90% of people could navigate to the FAQ section or had additional questions after reading, then the requirement and specification would not be met. Given more time, these steps for requirements 3 and 4 would have been completed.

For requirements 2-4, we concluded that the best way to determine whether or not the app is easy to use, easy to interpret, and has transparent calculations is to ask participants via surveys to get a better understanding of how they interact with the app. These tests would be empirical and would not be the most rigorous analysis due to limited time and representative sample bias. However, due to the subjective nature of the question being asked, we believe it would provide sufficient results. We have high confidence in the validity of these results, as the only way to get valid data about the user experience is to ask users. The contingency plan if these tests fail to produce believable results is to use our engineering intuition to make informed and intelligent decisions on behalf of our stakeholders. If these requirements and specifications are not met then the UI will need to be developed further.

The verification for requirement 5 was done during the user testing of requirements 2-4. During the user tests, we screen recorded the app, and measured the time between when the user presses the calculate button and when the calculation was completed. Although this test was less precise, we feel it is reasonable because initial testing showed the time to calculate results was nearly instantaneous, which is well under the 1 second requirement. Therefore, a precise value was not needed to show the requirement was met, so this coarser empirical measurement was sufficient. It was still a rigorous empirical test, and we were highly confident in the results of the test. Since the time to calculate results was well under the one second specification, the requirement and specification was met. Given more time, a timer function could have been implemented in the code in order to more precisely measure the time.

Validation Testing Results and Future Plans

Validation is an ongoing process, and was not completed within the project timeline. The validation process began during the concept selection stakeholder survey. In this survey, we found that 63% of respondents either would use or would consider using our app, and 77.8% of respondents would change their behaviors if a risk-assessment of high risk was output by the app [20]. However, the app will need further validation to ensure the original project problem is adequately addressed. To do this, we planned to conduct another stakeholder survey using the students from Professor Kaviany's class after they have used the beta version of the app. These students would be asked questions about if they would use the app, if they would change their behavior based upon the risk output, and if they would view this app as an accessible risk-assessment platform. Also, we planned to add an area for users to provide feedback in the final design, which can aid us in checking if the app is properly addressing the problem that we set out to solve. Lastly, we would like to have a discussion with the project sponsor to evaluate his perception of the designed solution to solve the problem. This discussion with the project sponsor will occur too late for us to make changes based upon his feedback due to his busy schedule, but the feedback can be used by future groups for further development and iteration if this project is carried into future semesters.

DISCUSSION

Several questions could have been explored and addressed given more time and resources in order to improve the process and outcome of this project. For the problem definition of the project, stakeholder interaction was very limited. Problem definition was guided mostly based upon stakeholder meetings with the project sponsor. Through the discourse in these meetings, the requirements and engineering specifications dictating the path of the project were developed. However, the problem may have been better defined, leading to better requirements, and therefore specifications, if more information would have been obtained from potential users of the app in addition to the information obtained from the project sponsor. For example, some questions that could have been asked to potential users as stakeholders are as follows: "What makes an app accessible or easy-to-use?" or "How would you define a high or low risk situation in terms of percentages?". This stakeholder engagement for problem definition could most easily be conducted through the use of a stakeholder survey. The reason this was not conducted for the project was that we initially thought the information and problem definition from our project sponsor was thorough, as well as the fact that obtaining a survey sample population of decent size that would have been representative of potential users would be extremely difficult and would have taken more time than we could afford to such a task.

In addition to increasing stakeholder engagement to improve the problem definition, further research could have been done into current models and platforms for these models before the project was delved into. Due to the limited time available, we instead primarily relied on the descriptions of these platforms from the project sponsor. Spending more time with these platforms, such as the Khan, Bush, and Bazant COVID Indoor Safety Guideline web application [7] or the Jimenez and Peng Aerosol Transmission Estimator Excel application [8], could have aided in us gaining an understanding of the problems with these existing platforms earlier and lead to better problem definition.

Similarly, due to limited time for the project, we relied primarily on information and resources given to us by the project sponsor during the design process. However, the design process could have been furthered by exploring more questions about the background of the problem and the project itself. For example , exploring some of the following questions more thoroughly could have led to a more comprehensive problem definition, understanding of the model, and a more comprehensive design process: “how do the assumptions made regarding particle size affect the accuracy of our model?”, “how can a spatial probability density function be incorporated into the model to go beyond the well mixed assumption more accurately?”, “how do mask filtration efficiencies differ between inhalation and exhalation, and how could this be implemented into the model?”, “would more discrete choices for user inputs lead to a higher degree of model accuracy and reduce uncertainty?”. Several assumptions related to these questions were made to simplify the design space for the sake of time. However, additional research into the medical parameters and further testing to quantify and analyze these questions could have improved the design process and project outcome. For example, a spatial probability density function distribution could be incorporated into the model to go beyond the well mixed assumption more accurately than the parameter uncertainties that are currently used. Additionally, and similar to how stakeholder engagement could be improved for problem definition, methods for user tests for concept selection and design verification could have included more participants from more diverse backgrounds. Multiple iterations of tests and design concepts used in tests could have also been used to see how users respond to improvements, and an exit survey to assess how users interpreted the results could have also benefited the design process.

The best strength of our design is the simplicity. It uses relatively basic input methods (such as discrete options and sliders) and simple qualitative questions that make it much easier for users to understand than other similar risk-assessment platforms, such as the web app or excel model. From this simplicity, a strength of the design is also the accessibility, as the need for technical knowledge is greatly reduced in order to use our design for risk-assessment as opposed to existing platforms. Another strength is the speed at which our design can be used, for both entering inputs and obtaining an output risk value. Other existing platforms also have a very fast computation time, but they take much longer to enter the input values in order to obtain the output risk value when compared to our design, which also aids in the accessibility of our design. Lastly, our model has a strength in predicting the risk for the user themselves. Some of the other platforms we have reviewed predict the time in a room for which there is substantial risk of any one person getting infected. This can be unclear as to how it pertains to a non-technical user, and therefore, our design presenting the risk for the user themselves aids in clarity and accessibility as well.

Our design does also have a few notable weaknesses. For one, the model has been reduced and assumptions have been made, such as neglecting the droplet settling speed and the effect of different aerosol sizes. Also, in order to simplify user inputs, the design uses some discrete categorical questions with corresponding discrete values, and these categories and values may not fit all scenarios for which users wish to assess their risks in. In addition, the design by default directly relates recirculation and filtration parameters to the input for ventilation type, as most users would not have a good method for estimating the recirculation and filtration setup in the room they are in. This choice was made both for simplicity and user accessibility. Although the user can choose to enter recirculation and filtration information if they are known, the default case means that some accuracy is lost. Due to these issues, the app is limited in accuracy for some situations, which is certainly a weakness. Another weakness is that our design slightly underestimates the risk compared to some other platforms, however, the risk is still within 5% of these platforms before accounting for uncertainties, so our risk output from the model is still accurate as discussed for verification testing. In addition, our design has a weakness as in that it has not been developed or tested for the visually impaired, or those with any other disabilities, and therefore may not be accessible to all demographics.

There are also a few possible improvements that can be made based upon this previous discussion of the problem definition, design process, and weaknesses of the design. As discussed, the problem definition, design process, and the design verification testing could be improved through the use of a more representative demographic pool and larger sample size in these processes. Another improvement could be made by further exploring other models that may work better for this application than the dose response model used in our design, since some models may possibly use parameters that are more compatible with qualitative input questions. Also, the design could be improved by further researching model parameter values for use in the design, as well as associated uncertainties with each model parameter value. In addition, the design could be improved by further developing it to be more compatible with the visually impaired, such as using large text or adding audio and dictation features. Another improvement that could be made would be to use the full model rather than the reduced model, and to incorporate the aerosol size distribution to improve accuracy.

REFLECTION

Our initial perspective when starting this project was centered around helping fight the spread of COVID-19 and helping people return to a more normal life without having to worry about infection.

Design Context

We talked primarily about the social context of this project, and how public health, safety, and welfare were the primary concern. The social contexts relevant to this project include public health concerns, quarantines, hospitalizations, and mortality rates associated with airborne diseases, which are all especially relevant due to the COVID-19 pandemic. This perspective has changed slighting over the course of the semester as CDC and University guidelines have loosened as a result of reduced case numbers and hospitalizations.

Initially the global application for this project was limitless. Since the design is a smartphone application, people all over the world would be able to download and use the product. This in turn would provide a much more significant opportunity to reduce the spread of COVID-19 by providing countless people with their real-time risk. The app did not make it onto the App Store due to unforeseen developer fees associated with Xcode, and as a result, the global context has been significantly reduced. However, the app remains ready to launch.

There are very few social impacts associated with manufacturing, using, and disposing of the product. There are no physical manufacturing, distribution, or disposal needs, and therefore none of the associated environmental or social costs. Instead, the environmental and social impacts include the mitigation strategies to reduce the risk of COVID-19 that may be increased to a change in perception of infection risk. Some of these mitigation strategies can be energy intensive, such as running HVAC systems in an effort to increase air changes per hour and decrease risk of infection. Additionally, waste management associated with increased mask usage and other preventative measures could increase as a result of this project in the case that users become more cautious after being aware of their infection risks. It is worth noting that although the end product is an app, the necessary computational power and the energy associated with running the app is minimal. This perspective has not changed since the beginning of the project.

The economic impacts associated with manufacturing, using, and disposing of the product are also not of significant concern. Originally, we believed that successful design and implementation of the app could have an economic impact, as it could aid the world socially and economically in the return to a “new normal”. However, over the course of the project our perspective changed and we realized that this goal was a bit naive. Additionally, the smartphone app was designed to be free and should have little to no development and upkeep costs. However, by the end of the semester we realized that a $99 annual fee would be required to keep the app on the App Store and an additional developer fee would be required to download the app on our phones for testing.

Several tools were utilized to characterize the potential societal impacts of our design. A stakeholder ecosystem map, shown in Fig. 2 on p. 6 was developed to show the relationship between stakeholders, the role they play in the project, and the potential impact they could have on the solution. Additionally, a stakeholder survey was conducted early in the project to determine what effect the realization of such a product would have on the target population. The results of the survey showed that a majority of the people surveyed would use an app that predicted the risk of covid infection and would change their behavior to mitigate that risk.

Inclusivity and Equity

Similarities and differences in culture, privilege, identity, and stylistic choices between team members both helped and hindered the design process. The four members of this team all share similar values and identities which can be convenient when confronted with a relatively straightforward problem like this one. For example, we found it very easy to work together and had similar ideas for the process and end goal of the project. However, these similarities also hindered our ability to create a more inclusive design. For example, the four of us all have iPhones and do not have any visual imparities that would affect our ability to use an app. As a result, we designed the app to run on iPhones and did not consider visual impairments when designing the interface. As a result, this means that some people would be unable to effectively use the app to assess their risk of COVID-19.

Additionally, similarities and differences in culture, privilege, identity, and stylistic choices with our sponsor both helped and hindered the design process. For example, our goals aligned with our mutual concern about COVID-19 infection and our desire to better the lives of those affected by the pandemic. However, there was an educational barrier to overcome in that Professor Capecelatro is significantly more knowledgeable in the fields of infection modeling and fluid dynamics. We were able to overcome this knowledge gap by reading relevant scientific literature, however, it took several weeks to get acquainted with the concepts which delayed the start of our design process.

The five most significant stakeholder groups identified during stakeholder analysis were the project sponsor (Professor Capecelatro), the project advisor (Professor Kaviany), the sampled population through stakeholder engagement, the general target users of the app, and ourselves (ME450 WN22 Team 11). In order to visualize the relationships between different stakeholders, a power vs. interest diagram was iteratively developed, as seen in Fig. 3 on p. 7.

The project sponsor, Professor Capecelatro, was regarded as a stakeholder with both high power and interest. Since the app uses models developed by Professor Capecelatro and his research team, he had both high interest and high power over the project. The project advisor, Professor Kaviany, was regarded as a high power stakeholder with a lower relative interest. Professor Kaviany had significant power and influence over the project and our design primarily through his high engagement and meetings twice per week. However, his interest was lower because he had less personal involvement with the project background itself.

The sampled population was a stakeholder group with high power and lower interest. This population had high power because they were involved in surveys and testing as a part of our stakeholder engagement plan, and the results from this engagement directly influenced our design choices. However, we believed these stakeholders were less interested in the project since they were asked to use the app as opposed to the general users who desire to use the app independently.

The general users are a stakeholder group with low power but high interest in the outcome of the project. This group is difficult to engage with, as it is a diverse and global population of anyone who is concerned with estimating their risk of airborne infections in any type of room. This is the group that the project is mainly aiming to serve. They had high interest because they seek out the app on their own.

Finally, as the engineers making the ultimate design choices, we (ME450 WN22 Team 11) were a stakeholder group with high power and high interest. None of the members in our team were overly concerned with COVID-19 direct infection. This is to say that none of the members of our team had compromised immune systems or fell into at-risk categories for hospitalizations or complications relating to COVID-19. As a result of our relative health and safety, we were not the target users for the type of app that we were developing. These interests conflict somewhat with the end users of our project since the end users will likely be people who are very concerned with their risk of COVID-19 infection due to preexisting conditions or at risk friends and family.

Additionally, the power dynamics between stakeholders should be considered, although no issues occured. It is important to acknowledge that compared to both Professor Capecelatro and Professor Kaviany, we are younger and less educated. Also, the general stakeholders will likely be constituted by many diverse lay people, so it is important to acknowledge that education, race, age, and more factors may impact the dynamics between the engineers and the users.

The main approach we used to include diverse viewpoints and ideas in our work was the use of stakeholder surveys. These surveys were used to collect information about how stakeholders would interact with the app, what stakeholders would like to see regarding the design of the app, and whether or not the app would change the behavior of those using it. Conflict between team members was essentially nonexistent and diverse viewpoints and ideas were discussed as a team and considered on a merit basis.

Ethics

The main ethical context of this project was the concern of miscommunication regarding the risk of airborne infection. Our personal and professional ethics within our team align with those of the project sponsor, project advisor, and the University of Michigan. In this specific context, there were two main ethical issues. One ethical issue is that if we communicate a risk to the user in the app that is below the actual risk, the user may make potentially harmful decisions for themselves or others based on the inaccurate information. Also, another ethical issue is if the app communicates a risk to the user that is higher than the actual risk, then we risk causing unnecessary stress and worry for individuals, and the users may even isolate themselves to avoid the higher perceived risk.

These ethical issues were managed by finding the balance between the most accurate room parameters possible and app inputs that are simple to understand and estimate. This is essentially the balance between accuracy and accessibility of our app and model. The app is not an effective solution to the problem if it is not accurate in its risk generation, but it is also not an effective solution if it is inaccessible. These values are in conflict because accuracy of the model depends on specificity of inputs, while more specific inputs might hinder accessibility by asking about information people do not know. The solution identified was to include accurate discrete categories for questions pertaining to room geometry and ventilation, while incorporating confidence intervals to account for any uncertainties associated with these discrete categories.

RECOMMENDATIONS

Improvements to the app will be primarily focused on making it more generally useful, both from a programming perspective and from a modeling perspective. From the programming perspective, more work needs to be done on the UI to ensure it works properly across all Apple Devices, which is a requirement for getting the app approved for distribution on the App Store. Some work has been done on this by using AutoLayout options for spacing text and input methods, but making the app work on extremely small screens will likely require making larger changes like adding scroll views. The app has also not been checked for accessibility guidelines like readability, so ensuring it meets those requirements is another possible design improvement. Additionally, making an Android version of the app would greatly increase accessibility.

From a modeling perspective, the app is currently hard coded with text and quanta generation rates for COVID, but the models used can be applied to any airborne disease. Making the text and parameters work for multiple diseases will also help make the app more applicable. Additionally, the current implementation of confidence intervals is generated using uncertainty for the inputs and not the probability density function described in the introduction. Although this was recommended by the project sponsor, a more sophisticated implementation of confidence intervals will also help improve the app. Adding an option to the final calculation screen where users can edit specific parameters to see how that changes the risk would also be a helpful addition. Finally, adding some of the design ideas to the presentation of the risk, such as color coding the results or adding comparisons, will help with interpretation of the results. This will require setting thresholds for high, low, and medium risk.

There are also some improvements that could be made to the parameter values and associated uncertainties. The parameter values could benefit from further research to improve accuracy of the model, as there are inconsistencies between literature with regards to these values. Also, the uncertainties associated with these values could be more formally researched and developed, since these uncertainties are currently estimates.

We also recommend further development of engineering specifications as well as more robust verification tests. Most verification tests relied on a small sample size that is not necessarily representative of the target demographic, so this is an important improvement to make for better verification testing to ensure that the design meets the required specifications.

Lastly, it is recommended that validation to ensure the design addresses the project problem is done using a version of the design published on the App Store in order to collect user data from a larger and more representative sample for the target demographic. This validation could most easily be achieved through a feedback option within the app interface, where users are asked about if the design is accessible and if it can be used to help inform behaviors.

CONCLUSIONS

The main goal of this project is to develop an app that is capable of providing their risk of infection with a confidence interval in an environment characterized by user inputs. Our app will move beyond the well mixed assumption and output an accurate probability range by utilizing the probability distribution of the spatial displacement of particles. This will be an improvement upon existing prediction tools since most of the current prediction tools do not provide confidence intervals and utilize the flawed well-mixed assumption.

We have been following the ME 450 Capstone Design Process throughout the semester and believe it sets us on the most complete design track as we move forward. Most stakeholders are the general public who are concerned about their risk of infection of various airborne diseases and aim to develop a product suitable for that audience. Outstanding stakeholders are the design team and our advisor and sponsor, Professor Kaviany and Professor Capecelatro, all holding more power over the development of the app than general users. We are carefully considering social, environmental, and ethical contexts throughout development. The app will be free to allow for the widest possible audience access to information about their risk of infection, as is our ethical responsibility. There is also an emphasis on the app being accurate, as we understand the consequences of unreliable conclusions if the model developed is not held to a high standard.

Five requirements and specifications have been determined that will allow us to track the development of our project. Each of the requirements was discussed at length during stakeholder meetings, and each of the specifications were chosen to provide the optimal benefit to our users. In addition, the feasibility of each specification was discussed in great detail to ensure that each one is achievable. The methods for validating the specifications are primarily user tests and surveys to understand and quantify how users are interacting with the app.

Since DR1, we have generated and evaluated various concepts related to the app. In particular, we have generated concepts in three main categories; different questions needed to obtain the user input to the models, different UI options for making those inputs, and different methods of presenting the concepts to aid in interpretation. A stakeholder survey evaluating the input questions has been completed, and it indicated that the proper direction for user inputs are qualitative as opposed to quantitative questions. The other concepts cannot be fully evaluated until a full release of the app is developed, as full user tests are required to determine which concepts best meet the engineering specifications.

In addition to these concepts, we developed an alpha build of the mathematical models in MATLAB in order to ensure we understand how to properly program the equations. The trends obtained from this model follow what one would expect from the equations, which indicates we have implemented the models correctly.

A sensitivity analysis has been conducted on the model parameters, and a relationship between the change in each parameter and the corresponding effect on risk output has been established. Through this analysis, the model has been reduced by neglecting the droplet settling speed, v_s, due to uncertainties associated with this parameter, its low sensitivity, and its negligible effect on the model. Also, neglecting settling speed leads to a more conservative risk assessment, as settling speed acts as a pathway for quanta removal. These reasons justify the reduction of the model in this way. The susceptibility factor, k, has also been combined with the pathogen production rate, P, and P will now be used as a quanta generation rate. This has been done to eliminate uncertainty in the model associated with the susceptibility factor, as well as because of the fact that more literature exists in terms of quanta generation rates than pathogen production rates.

A beta version of the app has been developed, with an interface and input questions reflecting our concept selection choices. This beta version incorporates qualitative questions as determined as the better option through the stakeholder survey for concept selection. Discrete parameter values have been incorporated for the inputs with categorical questions, while ranges or constant values have been determined for other questions. Optional questions have also been developed for users that know more about the room they are in. The main work left on the app is polishing the UI for all screen sizes and meeting accessibility guidelines so the app can be published. Refining the confidence intervals, making an Android version, and incorporating design ideas for results presentation are also improvements that can be made.

Verification and validation plans were developed. Verification tests have been conducted for multiple requirements. For Requirement 1. accuracy of model, the model output was compared to a previously verified Excel model for risk-assessment made by Jimenez and Peng [8], and all scenarios tested matched the Excel model within the confidence intervals. For Requirements 2 and 5, easy to use and quick runtime, user testing was conducted with the project advisor's, Professor Kaviany, ME 335 class. The users were able to enter the inputs quickly and the runtime was fast, but there were issues with the accuracy of the inputs, indicating further design refinement is needed. For Requirements 3-4, which are that the app is easy to interpret and transparent, a full distribution of the app is needed in order to obtain real world data on interpretation and transparency. Similarly, real world usage data and surveys of actual users are needed to complete validation of the app. As such, these parts of verification and validation will be completed once the app is published on the App Store.

ACKNOWLEDGEMENTS

First and foremost, we thank Professor Jesse Capecelatro for his project sponsorship, as well as his technical support, such as providing us with numerous references for models and outbreak data and answering our questions, throughout the duration of the project. We also thank our project advisor, Professor Massoud Kaviany, for his guiding hand through the project, his advice on how to improve the project, and his overall enthusiasm while working with us. Additionally, we thank Professor Kaviany’s ME 335 Section 002 students for their participation in user verification testing, as well as friends and family of the team for their support and responses during our initial stakeholder survey. This project would not have been possible without these people.

NOMENCLATURE

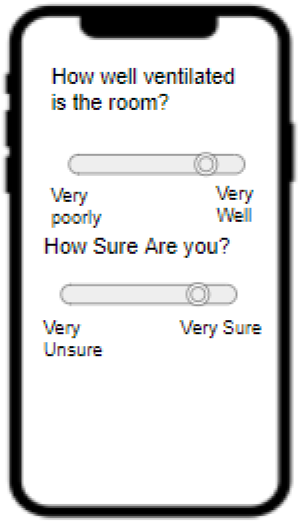
Parameter	Name	Value	Units
P_I	Probability of infection	0-1	N/A
N_I	Number of infected individuals	N/A	Count
S	Number of Susceptible individuals	N/A	Count
I	Number of infected individuals	N/A	Count
q	Quanta generation rate	1-1000	quanta/hr
Q_b	Pulmonary ventilation rate of a person	0.5-3	m^3/hr
t	Time	N/A	hr
Q	Ventilation rate	$1\text{-}10^5$	m^3/hr
V	Volume of enclosure	N/A	m^3
$C(t)$	Viral concentration	$10^{-4}\text{-}10^8$	pathogens/(m^3)
P	Pathogen production rate	$10^{-6}–10^9$	pathogens/hr
A	Footprint area of enclosure	N/A	m^2
p_f	Probability of droplet filtration via recirculation	0-1	N/A
Q_r	Recirculation airflow rate	$1\text{-}10^5$	m^3/hr
v_s	Droplet settling speed	$10^{-6}\text{-}100$	mm/s
λ_v	Virus deactivation rate	0-0.63	hr^{-1}
d	Virus dose	$10^{-4}\text{-}10^8$	pathogens
p_m	Probability of transmission through mask	0-1	N/A
k	Scaling factor of pathogens for infection	0.1-10	pathogens
X	Distribution Variable	$-\infty$ - ∞	N/A
μ	Mean of ln(X)	0	N/A
σ	Std. Deviation of ln(X)	0.9	N/A
$\overline{X}, X_{std}$	Mean, Std. Deviation of X	N/A	N/A
$\overline{C}, C_{std}$	Mean, Std. Deviation of C	N/A	N/A
$\beta_0, \beta_1, \beta_2, \beta_3$	Fitting parameters for C_{std}	-1.01, 0.11, 0.33, 0.10	N/A
λ_a	Air changes per hour (ACH)	1-6	hr^{-1}
λ_{ao}	Normalization factor for λ_a	1	hr^{-1}
H	Room Height	N/A	m
H_o	Normalization for H	1	m

APPENDICES

Appendix A - Subset of Generated Concepts

General Concepts:

1. Option to include social distancing (6 ft separation) in model analysis
2. Option to include vaccinations/boosters in model analysis
3. Option to include natural immunity in model analysis
4. Make it compatible with existing apps (e.g. ResponsiBLUE)
5. Convert for second function: The option to switch from COVID-19 to other airborne diseases should be implemented.
6. Link a google sheet in the FAQ/additional information section of the app so people can play around with the models and get a better sense of how parameters change probability of infection.
7. A history option could be implemented in the app so that users can refer to the output at a later time.
8. Allow people to save certain room parameters for frequently visited spaces (classrooms, etc)
9. Include statistics of local infectivity rates to estimate the number of infected people
10. Include help pop-ups that describe what each parameter is and how they are used
11. Include certain room types and pre populate v fields with typical values for those rooms
12. Share function for results (SMS, email, etc.)
13. All inputs to model are publicly available in the room itself so the user does not have to estimate i.e. a QR code the app scans that contains all the model inputs
14. Using sliders for both questions and confidence



15. Use pictures in order to visualize each input the users must enter
16. Provide a toolbar to separate sections of the app such as FAQs (could be on sides, top, or bottom)
17. Allow users to have space to choose units for inputs such as a dropdown box

18. Provide clickable choices as pre-sets for each input (i.e room size: small, office, medium, lecture room, large, auditorium)
19. Order inputs in terms of the model's sensitivity to each input [Heuristic: 38. Impose hierarchy on functions]
20. Create an interface that allows users to drag and drop different room parameters/qualities for an easy construction of the overall room characteristics [Heuristic: 8. Allow users to assemble]
21. Allow users to enter their acceptable risk level, and then give them room/environment parameters to use as filters to find an analogous room to their own [Trigger: 5. Subversion]
22. Create software as open-source software online so that the app and model can be further improved [Heuristic: 9. Allow user to customize]
23. Attach a sensor to the phone that can detect coronavirus particles or other proxy (CO_2, etc)
24. Use phone based measuring tool to find room dimensions include videos explaining how to find each parameter
25. Have a basic mode with more intuitive inputs and an advanced mode where the parameters are directly inputted
26. Interactive sliders with real-time calculations
27. Risk output comes with sensitivity analyses to show most important parameters that could reduce risk
28. Risk output comes with sensitivity analysis that shows which elements contributed most to uncertainty
29. Include a feedback form where users can ask questions or make suggestions
30. color-code the risk output green-yellow-red to help indicate what level of concern the user should have

User Input/Model Parameter Relationship Design Ideas:

31. Room Height
    a. Estimate actual height
    b. Qualitatively describe how tall the room is
32. Room Footprint Area
    a. Estimate length and width separately
    b. Estimate square footage
33. Room Volume
    a. Height x Area
    b. Height x length x width
34. # of people
    a. Visually estimate
    b. User square footage to estimate
35. Susceptibility
    a. Have you been vaccinated? Booster?
    b. Do you have natural antibodies? (have you had covid in the past 3 months?)
    c. What is your age?

    d. What is your fitness level?
    e. Are you immunocompromised

36. Breathing rate of others
    a. What activity is being performed? (speaking, singing, eating, sitting at rest)
37. Your breathing rate
    a. What activity is being performed? (speaking, singing, eating, sitting at rest)
38. Other people mask effectiveness
    a. What percentage of people are wearing masks
    b. What kind of masks are they wearing?
    c. How properly do you think the masks fit?
39. Your mask effectiveness
    a. What type of mask are you wearing and is it properly fitted?
    b. Or percentage of incoming particles blocked
40. Air flow rate
    a. Ask for it in V/T or in ACH
    b. Ask for ventilation type
    c. Ask qualitatively how well ventilated it is
41. Filter and recirculation
    a. Ask for raw parameters
    b. Ask if there is any recirculation or filtration
    c. Include in ventilation type
42. Time
    a. Time user expects to be in room?
    b. Time that is below acceptable risk level for user?
    c. Calculate for multiple time durations (i.e. 30 min, 1 hr)

User Interface Design Choices:

43. Sliders
    a. Room height
    b. Footprint area
    c. Room volume
    d. Number of people
44. Drag-and-Drop
    a.
45. Textboxes
    a.
46. Dropdown boxes/Pre-set clickable options (w/ visuals)
    a. Units
47. General Visuals
    a. Tutorial Videos
    b. Tutorial walkthroughs
    c. Visualize different app inputs/model parameters
    d. 3-D visual model of room as inputs are entered

Visualized Concepts for User Input Questions and User Interface Designs:

The following diagrams show COVID-19 vector tracking to represent how particles behave in a room with given parameters. The green arrows show particle removal by air flow, filtration, deactivation, and settling. The red arrow represents the quantity of particles that remains in the room. The phone on the left is a representation of what our app will look like with an example of a user input design choice. The blue arrow points from the phone to the parameter that is being assessed by the user.

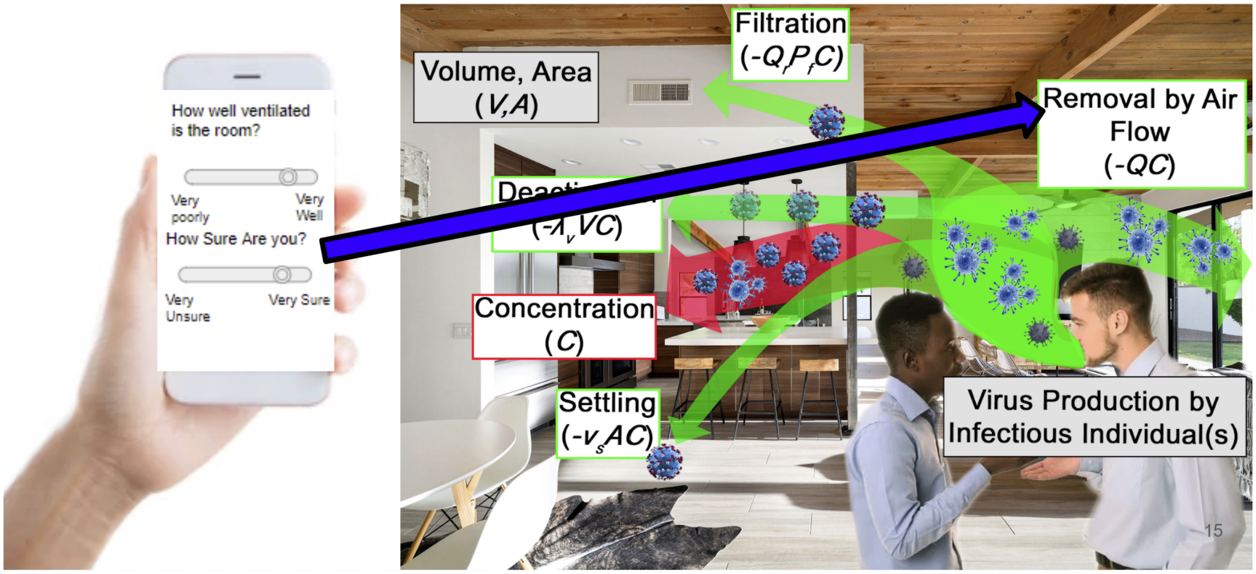


Figure A.1: A user interface concept consisting of sliders is shown. This allows for users to very quickly assess parameters and uncertainties by simply adjusting the slider. In this case the user is inputting ventilation information.

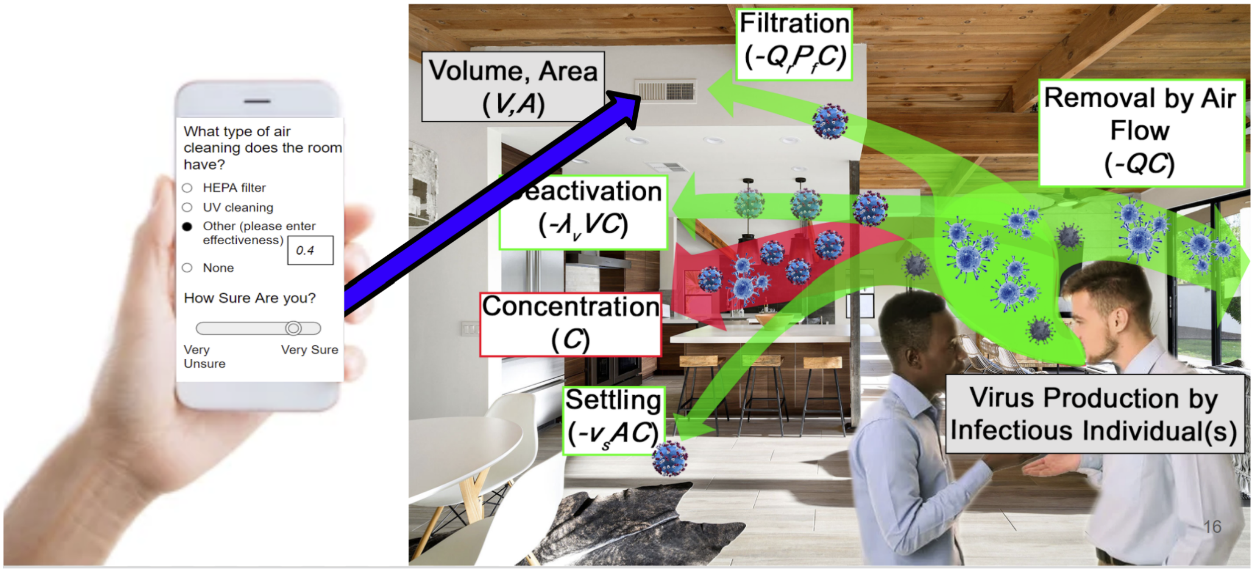


Figure A.2: A user interface consisting of preset options combined with an optional text box for added accuracy. Additionally, there is a slider to assess uncertainty. This allows for users to more accurately input parameters by predetermining exact values for the preset options. In this case the user is inputting filtration parameters.

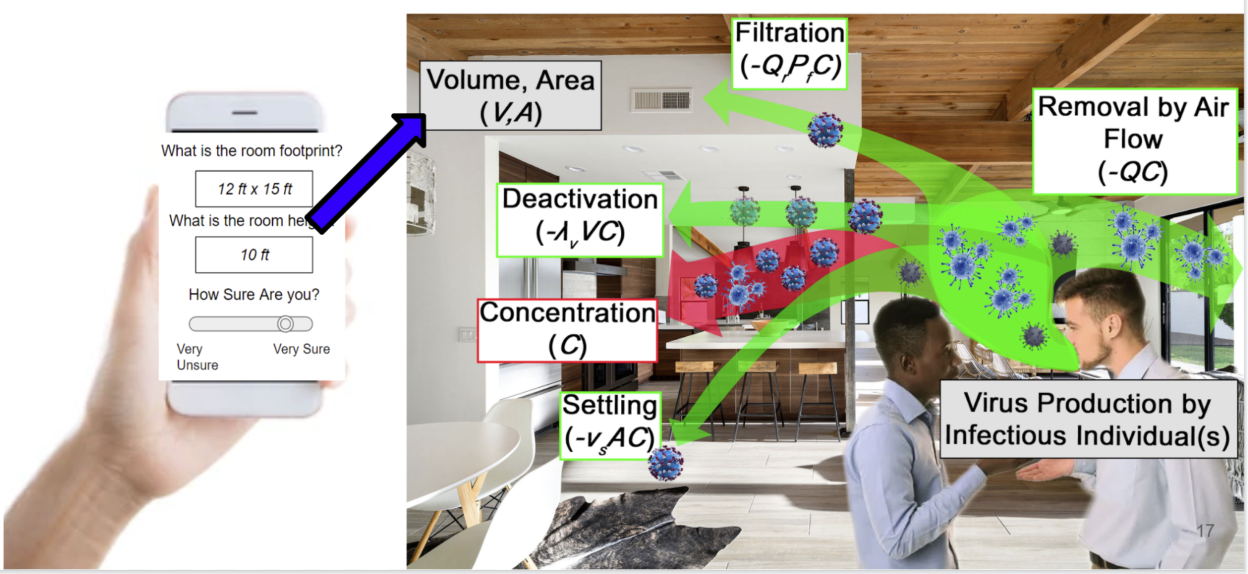


Figure A.3: A user interface consisting of text boxes is shown. Additionally, there is a slider to assess uncertainty. This allows for users to customize their surroundings with a high degree of accuracy. In this case, the user is inputting room volume.

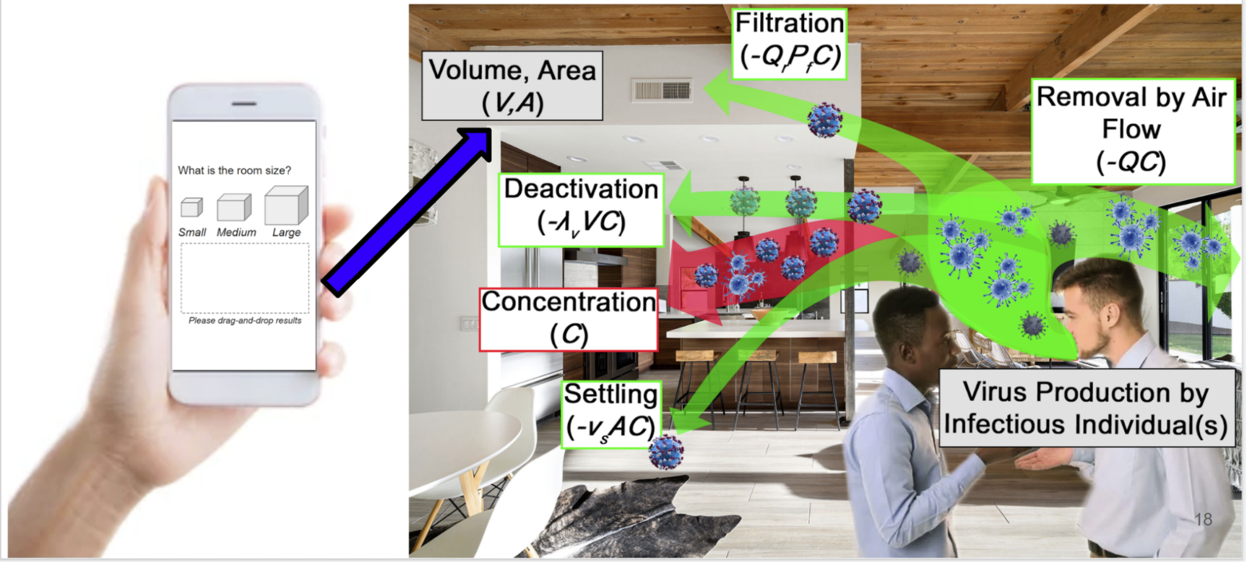
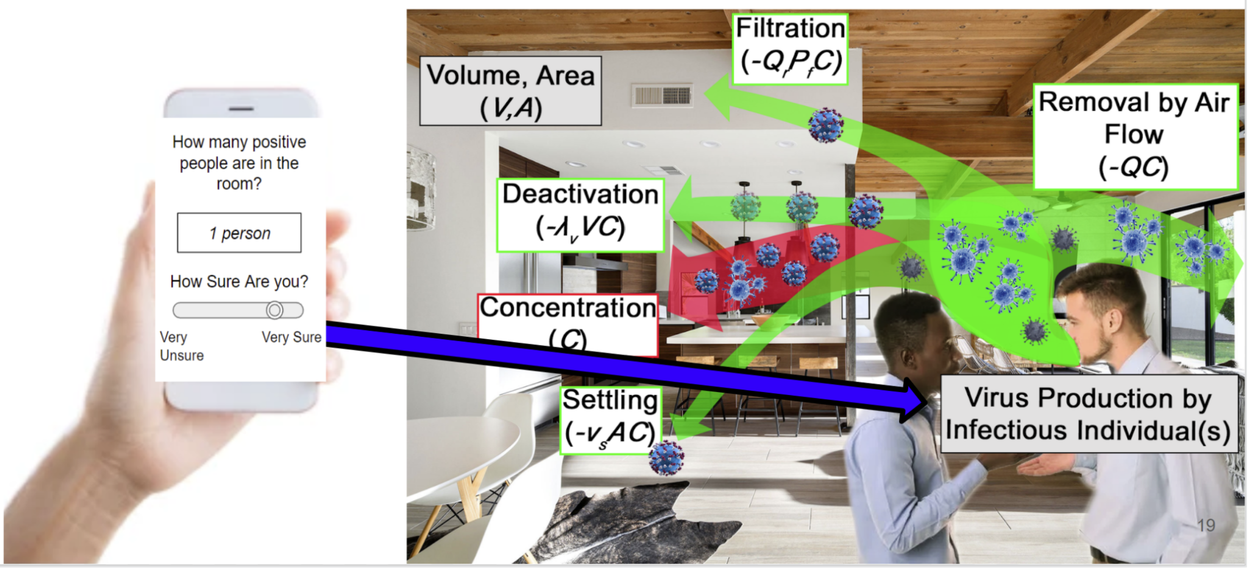


Figure A.4: A user interface consisting of drag and drop options is shown. This allows for users to very quickly enter inputs. In this case, users are inputting volume parameters.



How many positive people are in the room?

1 person

How Sure Are you?

Very Unsure — Very Sure

Figure A.5: A user interface consisting of text boxes is shown. Additionally, there is a slider to assess uncertainty. This allows for users to customize their surroundings with a high degree of accuracy. In this case, the user is inputting the number of infected individuals.

Appendix B - Default Model Parameters for Testing

In order to streamline the process of model testing, a set of default parameters was used. These parameters can be found in Table B.1.

Table B.1: Table showing the parameter values used for testing of the model

Model Parameter	Parameter Meaning	Default Value	Units
k	Susceptibility Factor	410	pathogens/quanta
I	Number of Infected	1	count
P	Pathogen Production Rate	50000	pathogens/hr
Q_b	Pulmonary Ventilation Rate	1.5	m^3/hr
p_m	Inhalation through Mask	0.5	N/A
V	Room Volume	50	m^3
Q	Ventilation Rate	100	m^3/hr
λ_v	Virus Deactivation Rate	0.63	hr^{-1}
p_f	Filtration by Recirculation	0.5	N/A
Q_r	Recirculation Airflow Rate	10	m^3/hr
v_s	Droplet Settling Speed	0.01	mm/s
A	Footprint Area	16	m^2

Please note that these were the default parameters that were determined and used prior to model reductions. After model reductions, the droplet settling speed, v_s, and the susceptibility factor, k, have been set to 0 and 1 respectively for all further tests. In addition, P has been changed to quanta generation rate and has much lower values. No default value has been determined for quanta generation rate since all tests after model reductions involved varying the quanta generation rate.

Appendix C - App UI Screenshots


Welcome! This app is a tool to help you estimate your risk of catching COVID-19. Please note the risk presented at the end is only an estimate, and its accurcy depends on the accuracy of you answers. To begin, please press the "Begin" button. If you want to know how your risk is calculate please press the "More Info" button.

This app calculates your risk using a version of the epidemiological dose-response model. It was developed by University of Michigan WN 2022 ME 450 Team 11 with assistance from Professors Massoud Kaviany and Jesse Capecelatro. A more detailed description of the model used by app is available in the Google Doc linked below:

Figure C.1: App Screenshots showing the welcome screen and background information screen

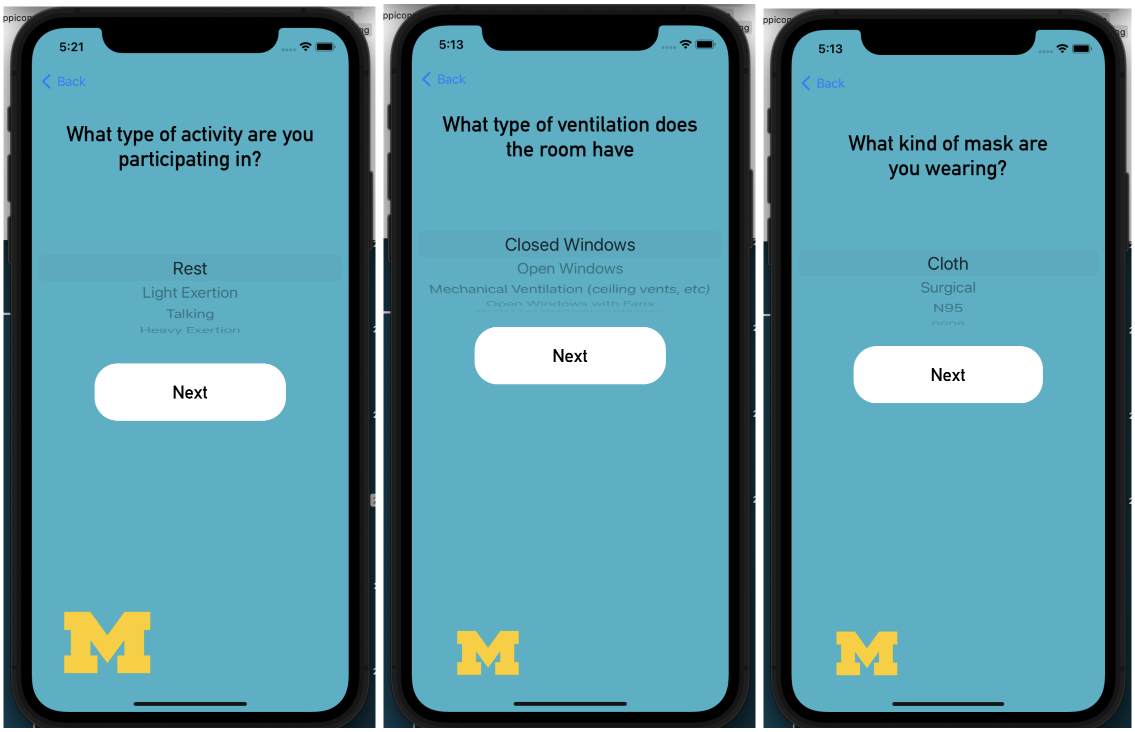

How many hours do you plan to be in the room?

How many people do you think are COVID positive in this room?

What type of activity are other people participating in?

Are you doing the same activity?

Figure C.2: App screenshots showing the questions for *t, I,* and *P.* Note the optional question asking if the activity is the same for Q_b.



What type of activity are you participating in?

What type of ventilation does the room have

What kind of mask are you wearing?

Figure C.3: App screenshots showing questions for Q_b, Q (and p_f and Q_r), and p_m.

How large is the room?

What is the footprint of this room? (ft²)

Your Risk of COVID-19 is:

0.021%

With Error bounds of

0.017% - 0.028%

Want to know how this risk was calculated? Press the "More Info" button below for some background information

Figure C.4: App screenshots showing questions for V and A, as well as the final results screen.

Appendix D - Model Parameter Uncertainty Values

Table D.1: Table showing the parameter uncertainties used for uncertainty in risk assessment

Parameter/Input	Value (if set)	Error (percentage or upper and lower bounds)	Direct or Inverse Relationship with Risk
Time	N/A	0%	N/A
I	N/A	0%	N/A
P (Rest)	3.1	1.6-4.6	Direct
P (Talking)	42	21-63	Direct
P (Light Exertion)	12	6 or 21	Direct
P (Heavy Exertion)	21	12-33	Direct
P (Singing Loudly)	270	135-400	Direct
Q_b (Rest)	0.54	0.27-0.81	Direct
Q_b (Light Exertion)	1.38	1-1.8	Direct
Q_b (Talking)	1.1	0.6 - 1.3	Direct
Q_b (Heavy Exertion)	3.3	1.8-4.0	Direct
Q (Closed Windows)	0.3	0.3-1	Inverse
Q (Open windows)	2	1-2.5	Inverse
Q (Mechanical)	3	2.5-4	Inverse
Q (Fans)	6	4-8	Inverse
Q (Better Mechanical)	8	6-10	Inverse
Q (Lab)	15	10-20	Inverse
Q (Bar)	9	5-13	Inverse
Q (Hospital)	18	12-24	Inverse
Q (Airplane)	24	18-32	Inverse
p_m (Cloth)	0.5	0.4-0.8	Direct
p_m (Surgical)	0.1	0.085-0.3	Direct
p_m (N95)	0.05	0.03-0.2	Direct
p_m (None)	1	1	Direct
Volume	N/A	25%	Inverse
Area	N/A	25%	Inverse
Height	N/A	25%	Inverse

Appendix E - README for App Final Design

```
#  README
This project folder contains the final code prototype for University of
Michigan ME 450 WN 2022 Team 11, the Real Time Risk Assessment of Airborne
Diseases
Team 11 consists of Jacob Keener, Andrew Hoelscher, Robert Faktor, and
Michael Scilkin, and was advised by Professor Massoud Kaviany.
The project was sponsored by Profesor Jesse Capecelatro

##Project Structure and files
The project consists of multiple view controllers, each of which contains
the code for one screen of the app.
The BasicXViewControllers each contain the code for getting the variable
X, and are all structured similarly
Most of the code in these files is copy and pasted, so most of the
comments are the first input BasicTimeViewController and only code that
does something new is commented on in later files (i.e. picker views in
Activity or sliders in Volume)
The ResultsViewController is the final screen of the app and presents the
risk calculation, the WelcomeViewController is the welcome screen, a dn
the InfoViewController gives a link to the design report background
sections
The inputParams file is how the model parameters are stored and where the
code for the calculations is used.
The Main.storyboard file controls the UI and determines how large each UI
element is, where it is located, its color, etc. To set up constraints on
UI elements, the Auto Layout tools were used.
The LaunchScreen.storyboard is the same for the initial loading screen
The assets folder can be used to add images to the app; the images folder
is simply a way to store them. The images need to be added to the assets
in order to display properly

##Storyboard structure and UI
The main structure of the storyboard is a Navigation Controller, which
allows transitions between screens.
For larger screens, a UI Scroll View can be used as seen in the advanced
mode view controller
Linking the UI elements to the code can be done by setting up the
storyboard view and the relevant view controller file side by side and
either control+dragging the UI element into the code (which creates a new
function or variable link) or by right clicking the UI element, selecting
the proper action, and dragging to existing code.
The code must have @IBOutlet or @IBAction to be interactive with the
storyboard
Segues, or transitions between apps, are made by right clicking the button
you want to activate the segue, selecting action, and dragging to the
desired screen.
If you want one button to segue to multiple screens under different
conditions, you must right click the view controller itself and drag to
the screens, name the segues different names in the identifier tab, and
then use the performSegue(withIdentifier: , sender: nil) function within
the proper button action with the condition checks
See BasicActivityViewController for an example
```

##ViewController Structure

In general, the view controllers are generated by right clicking the folder in the file browser on the left, selecting new file, and making a new Cocoa Touch class that is a subclass of UIViewController and written in Swift

The Cocoa Touch class ensures the proper library is important and the necessary functions are set up right.

The main required function is `override func viewDidLoad() {
        super.viewDidLoad()},`

which is where anything that needs to be initialized or processed when the screen is loaded is put.

The function `override func prepare(for segue: UIStoryboardSegue, sender: Any?) {}` which is how data is passed between views.

This is done by assigning the new view controller to a variable using `let varName = segue.destination as! NextViewController`

and then setting `varName.variable = self.variable`

Everything else can be structured as needed

##Common Errors or crashes

Some common errors with the storyboard include a SIGABRT, which usually means the next view controller does not have the same name as it was assigned in the prepare for segue function

To fix this, open the storyboard file and click on the view controller you are trying to transition to.

Check in the identifier tab on the right that the class name is the same as the one in the code and that the checkbox for inherit module from target is on.

The other common error I saw a lot was

>this class is not key value coding-compliant for the key

This usually means one of the @IBOutlets or @IBActions is not linked to anything, or is linked to an element in another view.

To fix this one, simply check all the associations of the UI elements in the view to check if they are linked to the wrong view, and make sure all @IBs have a filled in circle to the left.

Appendix F - Accuracy Verification Results

Table F.1: Table showing the results from verification testing for model accuracy compared to the Jimenez and Peng Excel model

Test #	Reduced Model Percent Risk	Excel Model Percent Risk	Percent Error
1	0.0689	0.07151	-3.6498
2	0.0014	0.00146	-4.1096
3	2.4085	2.45585	-1.9280
4	1.0123	1.07827	-6.1181
5	2.26E-04	0.00053	-5.73E+01
6	0.1355	0.1381	-1.8827
7	0.259	0.26734	-3.1196
8	0.0862	0.08917	-3.3307
9	0.0079	0.00815	-3.0675
10	0.5272	0.53683	-1.7939
11	0.064	0.06533	-2.0358
12	11.1362	11.51378	-3.2794
13	0.314	0.32763	-4.1602
	Average		**-7.3704**
	Average without outlier		**-3.2063**

REFERENCES

[1] Adams, WC, US EPA National Center for Environmental. 2009. "Measurement of Breathing Rate and Volume in Routinely Performed Daily Activities [Final Report]." WEB SITE. March 15, 2009. https://hero.epa.gov/hero/index.cfm/reference/details/reference_id/77086.

[2] Apple Inc. 2022 "What's Included - Apple Developer Program." Apple Developer. Accessed March 9, 2022. https://developer.apple.com/programs/whats-included/.

[3] Bazant, Martin Z., and John W. M. Bush. 2021. "A Guideline to Limit Indoor Airborne Transmission of COVID-19." *Proceedings of the National Academy of Sciences* 118 (17): e2018995118. https://doi.org/10.1073/pnas.2018995118.

[4] Birgand, Gabriel, Nathan Peiffer-Smadja, Sandra Fournier, Solen Kerneis, François-Xavier Lescure, and Jean-Christophe Lucet. 2020. "Airborne Contamination of COVID-19 in Hospitals: A Scoping Review of the Current Evidence." Preprint. Infectious Diseases (except HIV/AIDS). https://doi.org/10.1101/2020.09.09.20191213.

[5] Buonanno, G., L. Morawska, and L. Stabile. 2020. "Quantitative Assessment of the Risk of Airborne Transmission of SARS-CoV-2 Infection: Prospective and Retrospective Applications." *Environment International* 145 (December): 106112. https://doi.org/10.1016/j.envint.2020.106112.

[6] Capecelatro, Jesse. 2022. Group Interview with Professor Capecaltro Zoom.

[7] "COVID-19 Indoor Safety Guideline." n.d. Accessed February 2, 2022. https://indoor-covid-safety.herokuapp.com/.

[8] "COVID-19_Aerosol_Transmission_Estimator." n.d. Google Docs. Accessed February 2, 2022. https://docs.google.com/spreadsheets/d/16K1OQkLD4BjgBdO8ePj6ytf-RpPMlJ6aXFg3PrIQBbQ/edit?usp=gmail&usp=embed_facebook.

[9] Eisenstein, Michael. 2020. "What's Your Risk of Catching COVID? These Tools Help You to Find Out." *Nature* 589 (7840): 158–59. https://doi.org/10.1038/d41586-020-03637-y.

[10] Evans, Matthew J. 2020. "Avoiding COVID-19: Aerosol Guidelines." medRxiv. https://doi.org/10.1101/2020.05.21.20108894.

[11] Konda, Abhiteja, Abhinav Prakash, Gregory A. Moss, Michael Schmoldt, Gregory D. Grant, and Supratik Guha. 2020. "Response to Letters to the Editor on Aerosol Filtration Efficiency of Common Fabrics Used in Respiratory Cloth Masks: Revised and Expanded Results." *ACS Nano* 14 (9): 10764–70. https://doi.org/10.1021/acsnano.0c04897.

[12] Miller, Shelly L., William W Nazaroff, Jose L. Jimenez, Atze Boerstra, Giorgio Buonanno, Stephanie J. Dancer, Jarek Kurnitski, Linsey C. Marr, Lidia Morawska, and Catherine Noakes. 2021. "Transmission of SARS-CoV-2 by Inhalation of Respiratory Aerosol in the Skagit Valley Chorale Superspreading Event." *Indoor Air* 31 (2): 314–23. https://doi.org/10.1111/ina.12751.

[13] Ng, Simon. 2021 "Beginning IOS Programming with Swift and UIKit (IOS 15) - Sample." Accessed March 9, 2022. https://www.appcoda.com/swift.

[14] Noakes, C. J., P. A. Sleigh, A. R. Escombe, and C. B. Beggs. 2006. "Use of CFD Analysis in Modifying a TB Ward in Lima, Peru." *Indoor and Built Environment* 15 (1): 41–47. https://doi.org/10.1177/1420326X06062364.

[15] Peng, Z., A.L. Pineda Rojas, E. Kropff, W. Bahnfleth, G. Buonanno, S.J. Dancer, J. Kurnitski, et al. 2022. "Practical Indicators for Risk of Airborne Transmission in Shared Indoor Environments and Their Application to COVID-19 Outbreaks." *Environmental Science & Technology* 56 (2): 1125–37. https://doi.org/10.1021/acs.est.1c06531.

[16] Peng, Zhe, and Jose L. Jimenez. 2021. "Exhaled CO2 as a COVID-19 Infection Risk Proxy for Different Indoor Environments and Activities." *Environmental Science & Technology Letters* 8 (5): 392–97. https://doi.org/10.1021/acs.estlett.1c00183.

[17] Riley, E. C., G. Murphy, and R. L. Riley. 1978. "AIRBORNE SPREAD OF MEASLES IN A SUBURBAN ELEMENTARY SCHOOL." *American Journal of Epidemiology* 107 (5): 421–32. https://doi.org/10.1093/oxfordjournals.aje.a112560.

[18] Sze To, G. N., and C. Y. H. Chao. 2010. "Review and Comparison between the Wells–Riley and Dose-Response Approaches to Risk Assessment of Infectious Respiratory Diseases." *Indoor Air* 20 (1): 2–16. https://doi.org/10.1111/j.1600-0668.2009.00621.x.

[19] Tan, Sijian, Zhihang Zhang, Kevin Maki, Krzysztof J. Fidkowski, and Jesse Capecelatro. 2021. "Beyond Well-Mixed: A Simple Probabilistic Model of Airborne Disease Transmission in Indoor Spaces." medRxiv. https://doi.org/10.1101/2021.12.05.21267319.

[20] Team 11. 2022. "Stakeholder Surveys." Google Docs. Accessed April 6, 2022. https://docs.google.com/forms/d/e/1FAIpQLSekD6-u2ALXQgLlkBO0IQ--zsrEWlNBaDFDpBURHZIFuDV8mw/viewform?usp=drive_web&usp=embed_facebook.

[21] Watanabe, Toru, Timothy A. Bartrand, Mark H. Weir, Tatsuo Omura, and Charles N. Haas. 2010. "Development of a Dose-Response Model for SARS Coronavirus." *Risk Analysis: An Official Publication of the Society for Risk Analysis* 30 (7): 1129–38. https://doi.org/10.1111/j.1539-6924.2010.01427.x.

[22] Wynn, David, and John Clarkson. 2005. “Models of Designing.” In *Design Process Improvement: A Review of Current Practice*, edited by John Clarkson and Claudia Eckert, 34–59. London: Springer. https://doi.org/10.1007/978-1-84628-061-0_2.

[23] Zhang, Zhihang, Jesse Capecelatro, and Kevin Maki. 2021. “On the Utility of a Well-Mixed Model for Predicting Disease Transmission on an Urban Bus.” *AIP Advances* 11 (8): 085229. https://doi.org/10.1063/5.0061219.

TEAM BIOGRAPHIES

Robert Faktor (*rfaktor@umich.edu*)
Rob is originally from Westfield, New Jersey. At Westfield High School he was captain of the tennis team and was a part of the jazz band and concert band playing mainly tenor saxophone in both. He knew he wanted to study mechanical engineering because he wanted a scientific degree that would equip him to solve complex problems, but one that didn’t constrain him to a specific industry. At Michigan, he is a part of the Baja Racing team and active in the chess club. He is most interested in manufacturing systems and intends on pursuing a career in the field. He plays the tenor saxophone, piano, clarinet, and tuba, and is furthering his music education through a minor in music.

Andrew Hoelscher (*ahoelsch@umich.edu*)
Born and raised in Farmington Hills, Michigan, he was greatly shaped by this community and its public schools system. Throughout his academic career, STEM subjects have always been a passion, and he has always been a problem solver. This is why engineering is so appealing to him. He has been interested in aerospace, sustainable energy, automotive, and biomedical industries since the beginning of high school. Although he entered college interested in biomedical engineering, his direction shifted to mechanical engineering with a minor in Electrical Engineering, as he found his biggest interests are in Vehicular Research and Development and Sustainability. He is currently a DOD SMART Scholar and is committed to serving the public after graduation. He was also formerly a highschool all-american swimmer and now is an open-water swimmer who has swam the Straits of Mackinac.

Jacob Keener (*jbkeener@umich.edu*)
Jacob is originally from Ferndale, MI, which is where he gained my passion for engineering. He originally became exposed to engineering through our school district's FIRST Robotics Team, which he participated in from 7th grade until he graduated. He initially planned on studying aerospace engineering as he has a strong passion for the space sciences, but he felt that would restrict his career. Instead, he chose mechanical engineering as it is a broader field. He has since entered the Program in Sustainable Engineering, and hopes to somehow work in environmental engineering after graduating. At Ferndale, he also learned how to play the viola, a passion he continues to this day by pursuing a minor in music. He also likes to run and swim, and has participated in the club running team MRun on campus. Finally, on his longer breaks he likes to go on wilderness backpacking trips, and recently completed a 5-day solo through-hike of Pictured Rocks National Lakeshore in the Upper Peninsula of Michigan.

Michael Scilken (*mscilk@umich.edu*)
Michael is originally from Memphis, Tennessee. From an early age he was interested in engineering and physics. He graduated highschool with an international baccalaureate diploma and attended the University of Michigan after a gap year abroad. Freshman year of college he gained an appreciation for aviation and joined Air Force ROTC shortly after. He declared his major as mechanical engineering to get a broader understanding of engineering fields in general. After dozens of flight hours and several FAA tests and checkrides, he earned his private pilot license for single engine aircrafts. After graduation he will commission into the Air Force as a 2nd Lieutenant and attend pilot training at Columbus Air Force Base in Mississippi.