Allergy Prediction Using Artificial Intelligence

DESIGN DOCUMENT

Team Number: 48
Client: Ashraf Gaffar
Advisors: Ashraf Gaffar, Ashfaq Khokhar
Client Lead: Joseph Trembley
Team Lead: Noah Ross
Minute Taker: Ella Godfrey
Research Lead: Xerxes Tarman
Quality Assurance Lead: Alex Ong
Email: sdmay24-48@iastate.edu
Website: https://sdmay24-48.sd.ece.iastate.edu

Revised: 12/03/2023

Executive Summary

Development Standards & Practices Used

- AI / Machine Learning
- Cloud Development
- Version Control via Git
- Clean Code practices and maintainability
- User Interface design (Figma)
- Agile Development
- Sprint Planning - Determine next set of prioritized work or tickets

Summary of Requirements

- Model predicts allergic reaction accurately
- Model is deployed and available on a cloud platform
- Has clean user interface with wide-scale availability
- Simple to use: clients should be able to easily present data to the model and be returned a result
- Backend can send data to and from front-end

Applicable Courses from Iowa State University Curriculum

COM S/SE 309 - Software Development Practices: Comprehensive course in learning project management, working in a team, and using Git.

DS 301 - Applied Data Modeling and Predictive Analysis: Teaches many machine learning techniques which can be used to create the model.

COM S 319 - Construction of User Interfaces: Introduction to developing front-end user interfaces.

COM S 363 - Introduction to Database Management Systems: Teaches the fundamentals of databases and how to use them. Important for having a place to store and persist data for our model.

New Skills/Knowledge acquired that was not taught in courses

Working in a professional work environment that teaches agile workflow is beyond what formal education usually covers. We also needed to learn about AI as most of our group has minimal experience with AI development, and how to create and deploy a model using cloud platforms.

Table of Contents

Table of Figures

Table of Tables

1. Team

1.1. Team Members

Ella Godfrey, Joseph Trembley, Noah Ross, Xerxes Tarman, Alex Ong

1.2. Required Skill Sets for Your Project

AI modeling: understanding of how to begin building out and testing the model

front-end GUI: creating a user interface to interact with our model

Backend: sending data between front-end and model, and store within database

Database Management: storing data for the model

Communication (within a team and with a client): ensures high intra-team collaboration and understanding of requirements from client and advisor

Cloud computing: deploying front-end, backend, and model

1.3. Skill Sets covered by the Team

Ella has experience with front-end development working for a company that focused on web development.

Joseph is working on a minor in data science and has a basic understanding of machine learning/AI modeling. He also has 6 summers of interning with a Java development team working with backend applications and Amazon Web Services.

Noah Ross has 2 summers of experience developing User Interfaces on construction/agriculture products. His strengths include human/product interactions and defining customer wants/needs.

Xerxes has two summer internships working as an embedded software developer. He has experience collaborating and working on an agile team.

Alex has two summer internships with one carrying into the school year part time. Between the two internships, he contributed to both front-end and backend projects, giving a range of full-stack industry experience. He has also built personal and ecommerce full-stack websites as well.

1.4. Project Management Style Adopted by the team

As a team, we plan to be a waterfall+agile team splitting our work into different components, which can be worked on in order of importance while organizing the stages of components into stories we can all work on as needed. This will also provide us with the agile reflection step, where we can evaluate our progress and adjust accordingly.

1.5. Initial Project Management Roles

Noah: Team Organization, covers creating team meetings and anything to do team wise

Joseph: Client Organization, sets meetings with the client and handles all emails between the team and the client.

Ella: Minutes Taker, Records the minutes in a shared google doc.

Xerxes: Research, determines best technologies to complete the project

Alex: Quality Assurance, ensures all work meets requirements

2. Introduction

2.1. Problem Statement

The goal of this project is to predict whether a patient would have an allergic reaction to a medicine. It will use machine learning to make a decision based on factors about the individual along with the medicine itself. This will allow for faster and easier tests, as the model will have a rapid response time, and no additional patient information is required to run the model.

2.2. Requirements & Constraints

Model Requirements:

- Predicts whether a patient would have an allergic reaction to a medicine
- Able to process large number of input variables

Backend Requirements:

- Backend can send data from front-end to model
- Backend validates data and ensures it follows correct formatting
- Backend can return results to front-end
- Backend has limited read/write access to database

UI/front-end Requirements:

- Clear display of prediction and confidence level
- Location for user to upload information to test the model
- The remaining UI should be visibly appealing to the user
- Web accessibility from anywhere

Legal Requirements:

- Data collection and storage does not violate any health privacy laws

Testing Requirements:

- The model should be tested for an overall accuracy percentage to report

- Each component has multiple iterations of tests for any type of error
- System has tests covering entire scope of the project
- Logs should be implemented to catch faults or errors

Maintainability Requirements:

- File structure should be clear
- Code should be well documented

Data Requirements:

- Data stored in a database to be accessed by the model
- Database has security to prevent outside access

2.3. Engineering Standards

Since our project would take place in the medical field we should follow the IEC 62304 standard [1]. The IEC 62304 standard [1] specifies the life cycle for medical software devices such as risk management, software requirement analysis, software system testing, etc.

IEEE 11073 [2] is the "Health informatics - Point-of-care medical device communication," this provides a framework for compatibility between various medical devices and systems. It [2] defines communication protocols and data formats for exchanging information between medical devices and healthcare information systems, such as electronic health records or AI systems. When developing an AI model for allergy prediction, adherence to this standard can facilitate easy integration with other medical devices and systems, ensuring that relevant patient data can be efficiently shared and utilized for accurate predictions and informed healthcare decisions.

IEEE 2801 [3] is a standard built around the management of data in medical artificial intelligence. It [3] places a high emphasis on how data should be used and controlled once it has been gathered. Following this will help us ensure that we are using data in an ethical manner.

The model will be built using Python, because this language supports a wide variety of libraries, including ones with a focus on machine learning. One of the key libraries we will use is Keras, which is used to create a neural network system to train models.

2.4. Intended Users and Uses

Some of our intended users would be medical professionals, who could see if a patient might have an allergic reaction to a medicine; patients themselves, who can enter their information and determine if one of their medicines is causing allergic reactions; and medicine manufacturers, who can find whether their upcoming medicine could cause widespread allergic reactions.

3. Project Plan

3.1. Task Decomposition

- Predict whether a patient would have an allergic reaction to a medicine
    - Determine the tools and libraries to build out the AI prediction software
        - Deep Learning Libraries: Keras

    - Find an existing model since building one would take too long
    - Train the selected model with the data we find
    - Test if the accuracy (delta) is at the desired level, if not repeat the previous and current step
- Support processing of large input sets
    - Setup system environment in AWS
- Display prediction and confidence level clearly
    - Graphs and charts should be used to display basic numbers that are important to visually understand
        - Find a third-party graphing library / API to use that is compatible with our front-end language/framework
    - The media should be properly labeled and explained if necessary for transparency
- Location for user to upload information to test the model
    - User can easily fill out a form where they input the necessary information
    - Determine input file type (PNG, JPEG, etc)
- Create an application as an interface between our users and the AI model
    - front-end: React
        - Form which restricts user input to valid data
        - Deploy front-end to AWS
    - Backend: Lambda cloud functions written in Python for AWS
        - Create validation function for data
        - Function to send data between front and backend
        - Function to read/write from database
- The remaining UI should be visibly appealing to the user
    - Prototype a Figma design and mock-up
    - Define a color scheme to follow
    - Optimize loading time
- Collect data to train and test our model
    - Ensure data collection and storage does not violate any health privacy laws
    - Research which laws are relevant to our project
    - Collect data from advisor
- The model should be tested for an overall accuracy percentage to report
    - Define a percentage threshold to determine if our model is suitable for use
        - Use medical standards for effective diagnosis as a benchmark.
        - The threshold should be generous to start, then constrain it as we develop the project further
- Logs should be implemented to catch faults or errors
    - Logging tools implemented at front and back end
    - Errors give descriptive reasons as to what caused the error and remediation
    - Logs stored to a database or dedicated logging server such as Splunk
- File structure should be clear
    - Any files needed for the project are grouped together by how closely related they are
    - Data and code are in separate folders
- Code should be well documented
    - All necessary parts have descriptive comments
    - Detailed README file giving an overall view of how the project goes together

  - Possible UML diagrams to illustrate the data flow and end to end connections. Allows for easy high level understanding of our project for new users. (Sequence diagrams, class diagrams, etc)

3.2. Project Management/Tracking Procedures

Waterfall+agile project management best fits our project needs. Since many of the requirements can be developed independently of each other at first, it makes the most sense to work on them concurrently. It also keeps everyone working on something as new stories can be pulled in as needed. Using the waterfall style would force all of us onto one component at a time, and the work may not divide evenly between everyone. Waterfall alone also is missing the critical reflection step that agile has, meaning there is no real discussion for what is going well and what is not. Likewise, a fully agile management style uses an open-ended timeline and scope of the project whereas we have a plan for the tasks that need to be completed and when.

To handle version control and sprint planning, we can leverage resources such as Gitlab or Github. Any additions to the code base will be actioned by pull requests, where our team should request 1 review before merging to master to ensure code quality, best practices, and bug prevention.

Gitlab and Github also allow the option to create stories in order to ticket individual steps or features needed to be done to complete our project. An alternative could also be to use Trello, which is more dedicated to creating sprint boards and monitoring progress.

Discord and Text are our two main forms of communication. Quick updates and informal messages are sent through text, while group meetings and project-specific discussions take place within Discord.

3.3. Project Proposed Milestones, Metrics, and Evaluation Criteria

Data Milestones:

Get enough data to properly train the model (# rows/data points > 1000)

Determine appropriate prediction for dataset (Classification, Clustering, Regression, Ranking)

Establish data collection mechanisms (mysql and python)

Quality of data is established (null/invalid values < 5% of total data)

Model milestones:

Machine learning model is created (accuracy > 50%)

Model is refined (accuracy > 80%)

Model is finished (accuracy > 90%)

Model gives a rapid response to the user (time to complete request < 5 seconds)

Model handles a large number of input variables (possible factors in calculations >= 20)

User is able to simply input data through a user interface (less than 5 clicks to complete)

3.4. Project Timeline/Schedule

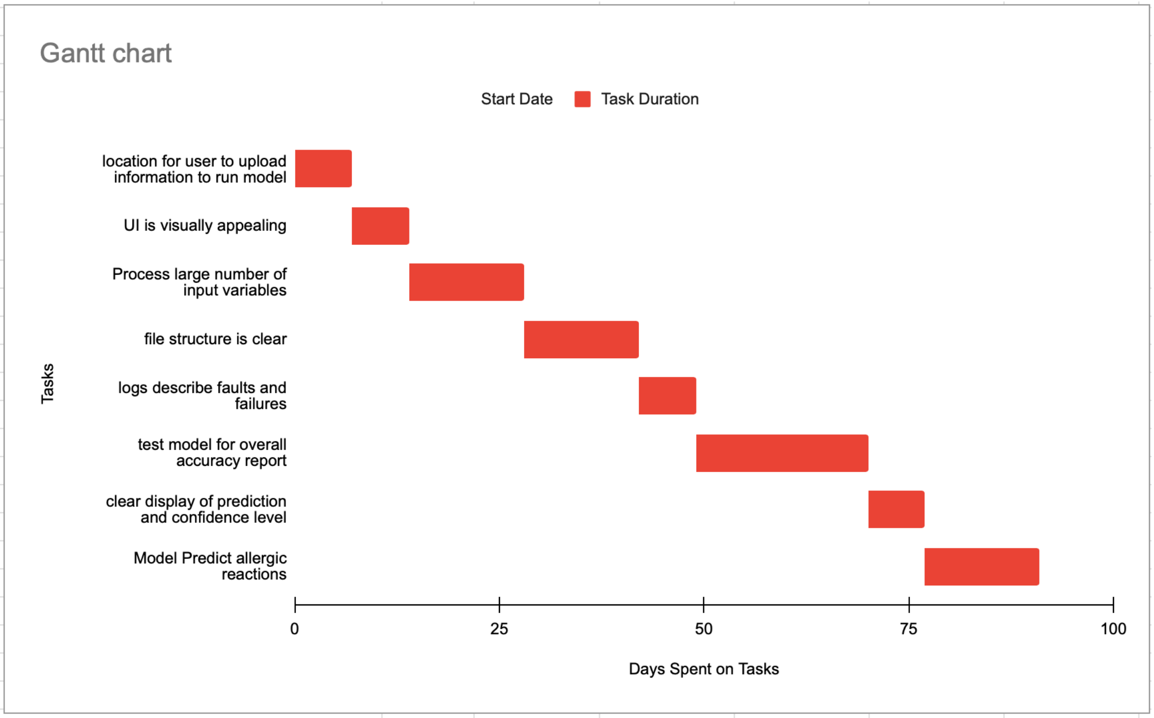


Figure 1: Gantt chart detailing the time to complete each major task.

3.5. Risks And Risk Management/Mitigation

Possible risks and probabilities:

- Model never reaches our minimum accuracy requirements (.7, there could be any number of reasons the model does not meet the goals such as a lack of a common link between the variables, or there is not enough data to properly train the model)
  - Mitigation technique: Compare AI model to others so that if a desired accuracy is never reached, we can explain why or suggest a different iteration for future attempts
- Front-end, backend, and model are unable to connect to each other (.2, even with a simple model, communication between all parts should be feasible without much difficulty)
  - Mitigation technique: Each system should be stress tested individually, but must also demonstrate the ability to effectively communicate with all other involved systems.
- User cannot upload/use their own information to get a result (.3, getting the user to upload information in the correct format may be tricky, but it should be possible)
  - Mitigation technique: We can test the import functionality with our own personal data and include major data importing formats to improve accessibility (such as .csv files)
- Error handling/logging not implemented (.2, most error handling and logging will be built alongside the major components of the project, so the risk of this problem occurring is low)

    - Mitigation technique: Each system should log individually to provide a transparent view of any errors or failed connections to other systems
- Code structure is unclear (.05, any system which divides the front-end, backend, model, and data into separate components will suffice the minimum requirements. Even if this is not followed, the requirement should not impact the model's functionality.)
    - Mitigation technique: Coding structure must be simple and clear, and any formatting should be consistent and listed in the README portion of the project

3.6. Personnel Effort Requirements

Task	Required person-hours	Explanation
Model predict allergic reaction	20	Since none of us were very familiar with machine learning going into the project, we will need to ensure we understand the process along with implementing it.
Process large number of input variables	12	While most of this is done while developing the model, ensuring that the input variables are all being used will take time.
Clear display of prediction and confidence level	3	This requires accessing the model to get these pieces of information along with creating components in the display for these parts.
Location for user to upload information to run model	4	The task is not too complex and should not require much work. Time also includes time to test the upload process and ensure everything works.
UI is visually appealing	2-4	The time we could spend on this should vary depending on how much time we have, as it is less important than the functionality of the project.
Collect data to train the model	2-8	The time this takes can vary greatly, as it will be difficult to source legal and ethical data if such data is not provided for us

Completed within 2 semesters	N/A	Since this is a resource constraint, it does not have hours dedicated to it.
Test model for overall accuracy report	30	Testing and retraining the model depending on our needs may take the longest amount of time, as it may involve switching to a new type of model if the one we use does not work.
Logs describe faults and failures	2	In order for us to understand whether the process is happening as it is supposed to, we will likely need logs along the way, but we should spend some time at the end making sure all possible outputs have log output and can be traced or interpreted.
File structure is clear	1-2	This should be planned at the beginning so that the code does not have to change structure much, but should not take much time.
Code is well-documented	2	This should be done as code is created, but allocating time to clean up and explain the repo at the end if there is time would be beneficial.

Table 1: Table detailing each major task, time to complete in hours, and explanation for how we estimated the times.

3.7. Other Resource Requirements

Computing power: since the machine learning model will be a complex model, we may need to utilize additional computing power beyond what we are able to produce ourselves.

Datasets: to train and test our model we will need a large dataset with basic medical information to ensure any factors can be accounted for by the model.

Monetary: if we are not careful, it is easy to incur a large charge when using the cloud platforms to store and run our model, which could affect how often we run and test the model.

4. Design

4.1. Design Content

Since this project is a software-based project, the design content will relate to the flow of data and functional components instead of physical hardware. The design will have a front-end that the user interacts with, a model which makes the prediction, and the backend to connect the two together.

4.2. Design Complexity

Our final design consists of three major components: the front-end, the backend, and the model. The front-end will be the interface seen directly by the user and requires a careful design to ensure users can navigate it properly. The backend will be built using cloud functions to call off to the model and return the result to the front-end. The model will create a prediction for whether the patient has an allergy based on the input data received. Some of the applicable principles are keeping the model and its data secure, as well as creating a transparent view of how data is used. Developing an ML model for our project presents a significant challenge as we strive to maintain high accuracy. Since the model will be trained with a large dataset with many varying input types, there may be numerous iterations before one begins to work as intended. Additionally, it is crucial to create a robust testing suite to safeguard against overfitting the model. Since the model will be hosted on a cloud-based platform, we will need to ensure that it can be accessed from anyone without compromising the integrity of the data to follow artificial intelligence engineering principles.

4.3. Modern Engineering Tools

- Model Creation: Keras API -
  - Keras serves as a high-level API for the TensorFlow machine learning library, known for its emphasis on providing a fast and user-friendly interface for creating production-ready ML models. Therefore, our group will be using this library to create our prediction model.
- Model Hosting: Amazon Web Services -
  - Since our model needs to be hosted on a cloud service, we looked into both Google Cloud Platform and Amazon Web Services to train our model, and we settled on using Amazon Web Services as our primary provider. A cloud platform lets us make use of tools designed for this type of wide-scale machine learning development and host our components in a combined manner.
- Model Workbook: Jupyter Notebook -
  - Jupyter Notebook is a browser-based tool for developing and presenting data science projects. It is specifically designed to visualize and execute Python code, which is useful in developing machine learning models. Our group will use Jupyter Notebook as our primary development tool for our machine learning model. Like the backend, both of our cloud services have options to host Jupyter notebooks natively. Amazon Web Services has Jupyter integrated within SageMaker, so no additional configuration is required to use this.
- Backend framework: Serverless functions -
  - In order to keep our backend resilient, we will be building serverless functions utilizing the cloud service as the main component of our backend. Each of the

cloud services has their own method of creating this. Amazon Web Services has Lambda, which can quickly run the functions required to transfer the data.

- Front-end framework: React.js -
    - React.js is a front-end JavaScript framework designed for building responsive single-page web applications. We will be using this library to develop the user interface for inputting data and displaying results from our ML model.

4.4. Design Context

Area	Description	Examples
Public health, safety, and welfare	Our project aims to use AI for allergy prediction. The main community area affected by our design will be the Public health, safety, and welfare area. By creating a tool that allows medical professionals to input patient data and symptoms, the goal of our design is to be able to predict specific allergies a patient may have. This can help provide faster and more accurate results, while reducing human error. This can lead to better treatment that benefits both the patient and medical professional.	Increase personal awareness of possible allergies.
Global, cultural, and social	With our project relying heavily on AI and sensitive patient information, we need to ensure that data is secure and that our design follows HIPAA regulations as this can impact the Global, Cultural, and Social areas. With our data being ported to a third party library with Keras, we need to ensure data does not get leaked onto the public web.	Keep user data secure, and follow HIPAA regulations.
Environmental	There is not much to consider outside of the energy and resources we need to consume to run our servers and AI model. Since most of our services will be through the Cloud, the main environmental considerations would be how much resources and energy is required to run our host machines.	Restrict computing power and energy consumption.
Economic	If our product is used by real medical professionals and facilities, an beneficial economic impact could be the resources, time, and overall cost saved by being able to predict patient allergies faster and more accurately.	The model prevents the need for expensive allergy testing kits. AWS provides free tiers which should suit all of our request needs during development.

	This would allow doctors to spend less time having to come up with the diagnosis and reduce the chance of inaccurate diagnoses. For our organization, some of the economic impacts to consider for us would be the third party sources we use and if we need to cover any costs for the operations if they were to exceed the free plans. Some examples are that cloud services such as AWS support free services up to a certain amount of bandwidth / requests. The same could be said for the Keras libraries and is something we may have to consider if our design scales. For now, as a small isolated school project, we believe there shouldn't be many financial impacts to consider as our project should not have to scale to a point which exceeds any of the free plan resources.	

Table 2: Table describing the impact our project may have within various areas of concern.

4.5. Prior Work/Solutions

Allergen testing is already done, but it is mostly done on test animals and in some cases humans. Monitoring skin pricks or digestive reactions to new products in animals like guinea pigs and humans gives us most of our current insight into allergic reactions. Not much exists in predicting allergic reactions like we are trying to do.

It is only in recent years where the idea of using artificial intelligence to predict any sort of medical event has become popular. While some of these models exist, none seem to have gained a large-scale adoption in the medical field. An article [4] published in August 2023 states that even though artificial intelligence is not at the point where it can currently be used in the industry, it will approach that point and is an overall benefit to healthcare.

We are not knowingly following in the footsteps of any other projects or programs as Machine Learning is pretty new, and has yet to be widely adopted in the medical field.

The article [4] states that the biggest hurdles to overcome with creating an AI model that could be used in the healthcare industry are the wide scale data requirements, lack of ability to see how the model processes data, and difficulty in validating the model. In order for a machine learning model to begin predicting data correctly, it will need a significant amount of data.

4.6. Design Decisions

One of the major design decisions we needed to make in the project was choosing whether to use Amazon Web Services or Google Cloud to use computing power and host our machine learning model. To make this decision, we had to carefully compare the features offered by both such as pricing, ease of use, and functional capabilities. After comparing them both by seeing available resources and using our own experiences with them, we elected to choose Amazon Web Services as our primary cloud platform for the project.

Another major decision we made was choosing which language to build our front-end with. We chose to use React, since many of us have experience working with the language, and its flexibility will be a great benefit to us as we implement our design.

The last major decision for the project was choosing how to create our backend. After some thoughtful discussion, we decided that we wanted to use a cloud function which would call our model and return the data back to the front-end. This method requires less coding overall, and it allows us to build the backend using the same cloud platform as the machine learning model uses.

4.7. Proposed Design

4.7.1. Design 0 (Initial Design)

Design Visual and Description

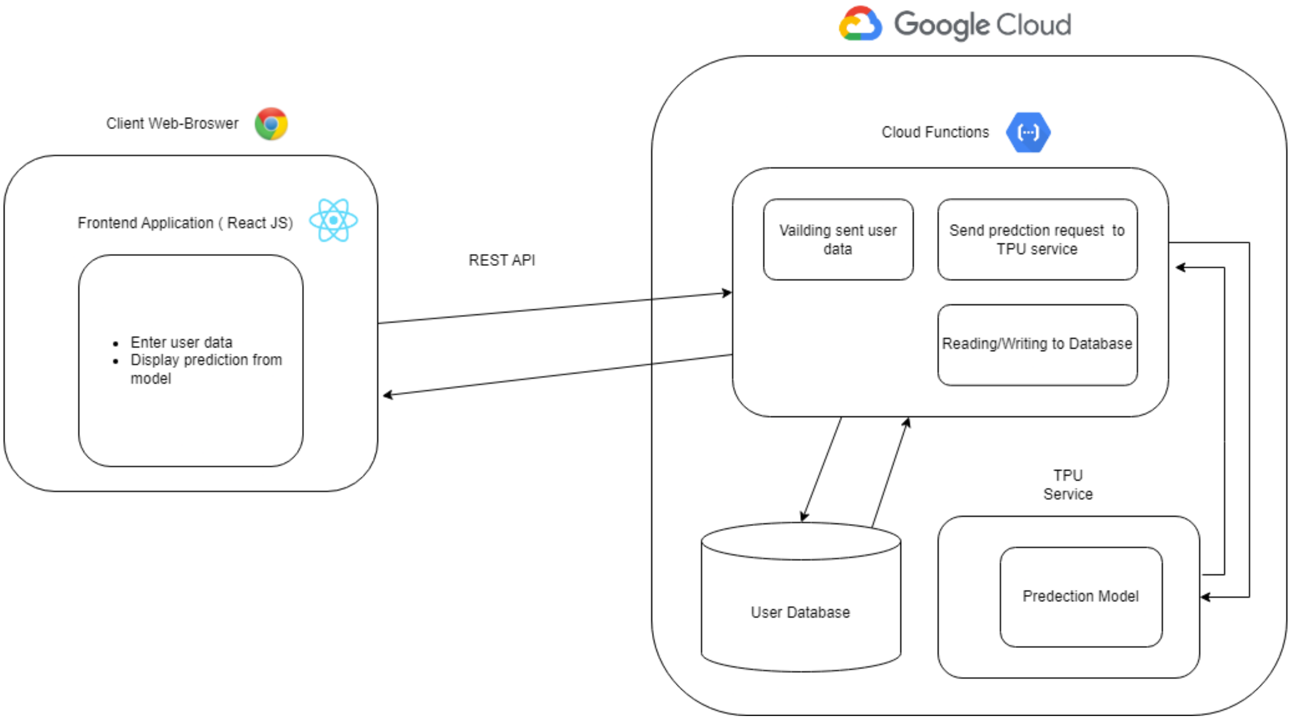


Figure 2: Initial design of the project with the React front-end, a backend utilizing cloud functions, and the prediction model.

The first portion of the design is the React front-end. It allows the user to enter the data and display the prediction. This component does not need much itself but needs to connect to the backend, which is possible with the given design. Since both the cloud function and model will be using the same cloud platform for their operations, they are placed under the same category. We used Google Cloud as a cloud provider option, but have since chosen AWS. Within the cloud platform, we have the cloud functions, the database, and the TPU service, a circuit designed for neural networks.

Functionality

In the current design, the user enters data through the front-end to get a response. The data is passed into the cloud, entering the cloud function, where the data is added to the database and sent to the model. The model, which at this point has already been trained, looks at the data and makes a prediction. The model sends its prediction to the cloud function. The function sends the data back to the front-end and adds the prediction to the database where the data is stored. The front-end displays the prediction to the user.

4.7.2. Design 1 (Design Iteration)

Design Visual and Description

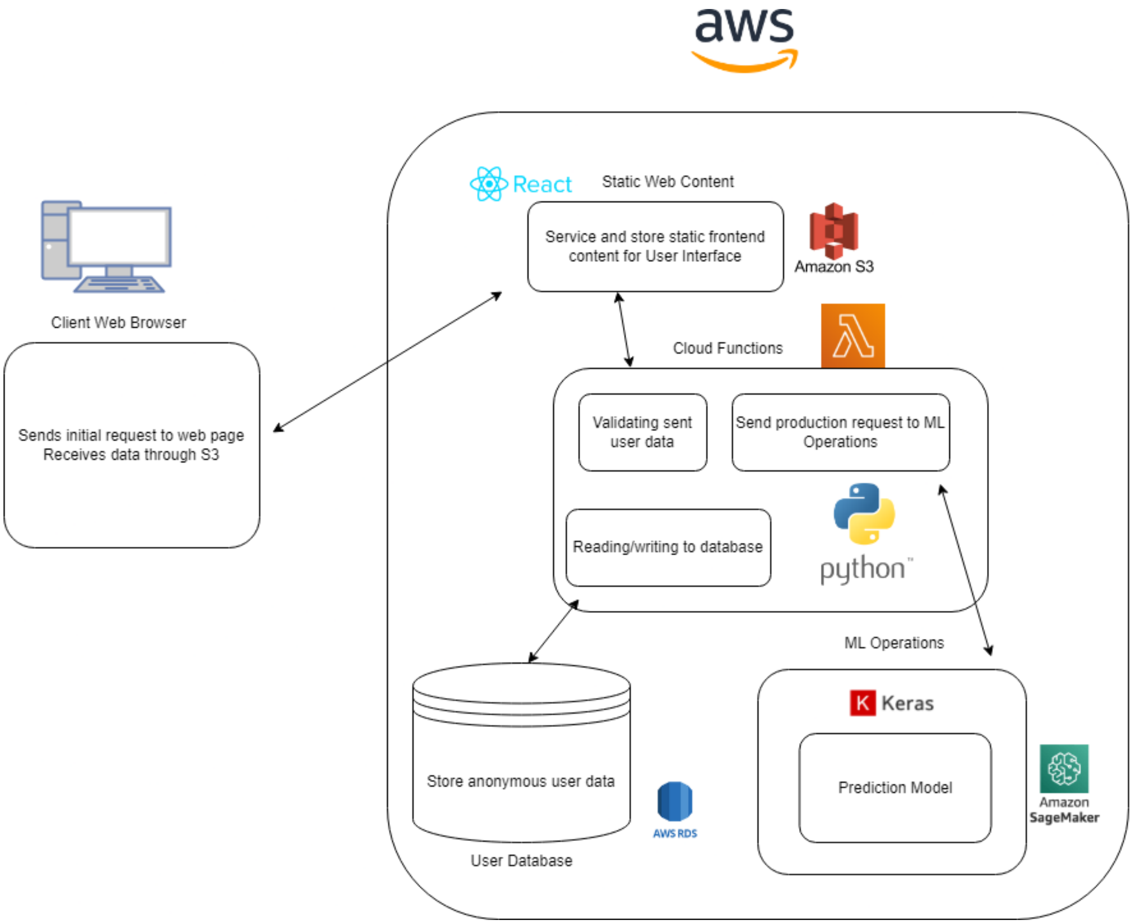

aws
React
Static Web Content
Service and store static frontend content for User Interface
Amazon S3
Client Web Browser
Sends initial request to web page
Receives data through S3
Cloud Functions
Validating sent user data
Send production request to ML Operations
Reading/writing to database
python
ML Operations
Keras
Prediction Model
Amazon SageMaker
Store anonymous user data
AWS RDS
User Database

Figure 3: Revised design of the project, including specific services within AWS to be used in the project.

Front-End (User Interface)

The user interface of our design will be a desktop web application. The user interface will be responsible for presenting a user-friendly interface for entering patient information along with relevant predictors. Additionally, it will facilitate the input and listing of chemical components to be sent to the allergy prediction model. Additionally, the front-end will handle the display of the prediction results generated by the model.

Backend

The back end will play a crucial role in managing requests from the user-facing web application. This involves updating a database to store patients' information and making prediction requests. Our backend architecture will comprise multiple AWS services. The front-end will be stored and delivered through an S3 bucket. The API, managed by AWS Cloud Functions, will be responsible for forwarding protection requests to our ML model, authenticating users, validating user input, and handling database queries for storing user and patient information. Additionally, we plan to host our prediction model using AWS SageMaker, ensuring accessibility through our Cloud Function.

4.8. Technology Considerations

By automating the technical aspects typically handled by a medical professional, our project has the potential to significantly decrease costs associated with drug prescription and development. However, a major challenge we face is our heavy reliance on obtaining high-quality data for our predictions.

One drawback associated with employing a machine learning model for allergic prediction is its reliance on the quality of the training data. The model may encounter challenges with chemical components that are not well-represented in our training dataset.

4.9. Design Analysis

As our understanding of what we needed for the project changed, we had to make some revisions to our design. Once we elected to choose AWS as our primary cloud service, we needed to change the design to reflect this decision and to illustrate which services would fulfill which purpose within the context of the project as a whole.

We have tested hosting each component individually on AWS. The front-end can be reached using a web browser and can make a call to the backend, receiving a response. In addition, the model can be stored on AWS.

Furthermore, during the round-trip testing phase, we initially decided to utilize Java for the backend functionality on AWS. However, after developing a function, we observed that Java development was challenging to integrate seamlessly with AWS. Subsequently, we discovered that developing with JavaScript proved to be a much smoother and more efficient process.

5. Testing

For the different areas of the project, there will be individual tests that ensure each section of the component works and fails when intended, and the system as a whole will be tested. One of the unique challenges with this design is testing the functionality of the AI model rather than testing just the accuracy. Testing the accuracy is straightforward, as the model is run on test data and evaluated.

5.1. Unit Testing

AI Model: We will need to test the AI model for accuracy separate from the other components. We will have a baseline model that provides a reference point for comparing the performance of the model. Looking through the provided errors will show the weakness within allowing us to focus on the weak parts of the model. The data set will be split into two parts where one part can be used for training and the other used for testing. In addition to testing the accuracy of the model, we will test whether the code for the model does what is intended. We will integrate our model testing into a CI/CD pipeline.

Individual Cloud Functions: Our backend cloud functions will each need to be tested individually. This will include validating data sent by the user and data parsing for the response. Our back-end cloud functions will be tested using the PostMan API. For each individual cloud function, there will be a corresponding unit test that sends a request and validates the response, along with unit tests that ensure failures and errors are correctly handled by the cloud functions. In most Python scenarios, we can use the unit test framework to run and validate our tests.

Basic front-end component testing: Our front-end will be made up of components such as buttons, text input fields, etc. We should test the behavior similar to how a user would use the application using libraries such as the React Testing Library (RTL). This library is useful for asserting state or values from front-end components such as text entries, if a button is disabled or enabled, etc. We will also use Jest to run the tests, and confirm whether they failed or succeeded.

5.2. Interface Testing

The front-end interface of our project is a front-end built with React. The React Testing Library will be useful in writing the interface tests by creating a virtual DOM for the tests to run in. We will be running the tests Jest, a JavaScript testing library that is known for its simplicity and isolated tests. Isolating our tests lets each component of the interface be tested without the influence of another.

The backend interface will be built using cloud functions written in Python. Similar to the unit testing stage, the unit test library should provide what we need to test the entire backend suite.

5.3. Integration Testing

One of the major integration paths we will need to test is connecting the front-end to and from our cloud platform. This will involve careful planning from both sides to ensure it works as intended. To ensure that the front-end is receiving calls, we can mimic a response that uses the same path as a real response would. Much like the interface testing, this can be done using Jest.

Our backend cloud functions also have to make calls to both our database and our AI model. We should test these together to ensure we can read/write data to a mock database and send mock calls to the AI model. This testing framework, mock from unit test, is specifically designed for the mock calls that trigger the cloud functions.

5.4. System Testing

To test the entire system we will need to make sure we have a list of all the requirements our project should meet. This will allow us to create a test plan that outlines the key functionalities. We will continue to use unit tests in our case Jest to make sure they meet their specifications. For integration testing, we will use end-to-end testing to test user interactions. For backend we can use Postman to validate the communication between the front-end and backend. We can also do performance testing that can test response time. This should also be continuously monitored for security, performance, and user experience.

5.5. Regression Testing

We will build new components and test them with the old components before merging them to main. A CI pipeline will be established to automatically run these tests for each new branch looking to merge. By keeping new components and features in separate branches, we can mitigate risk associated with regressing or damaging finished features. On the ML side, since our model will undergo multiple training iterations on the same data, we aim to prevent overfitting by employing a subset of the data for testing. This allows us to assess whether the model is overfitting or not. Our strategy to prevent overfitting will be a notebook with k-fold cross-validation. This enables automatic assessment of whether the model is improving or succumbing to overfitting for our dataset. Moreover, this method will be useful in the beginning of our project for selecting the ML approach that best suits our data.

5.6. Acceptance Testing

To demonstrate that the design requirements are being met we can establish clear traceability between the design requirements and the testing phases. This can be done by having each requirement aligned with a list of test cases which will ensure that every requirement is tested.

To involve the client in the acceptance testing we will have them test the project to see if it works as they would expect it to. The client can provide feedback during this process which can range from defects, deviation within the requirements, or places where the software doesn't meet expectations. This will also ensure that both functional and non-functional requirements have been met and that the final product aligns with the expectations.

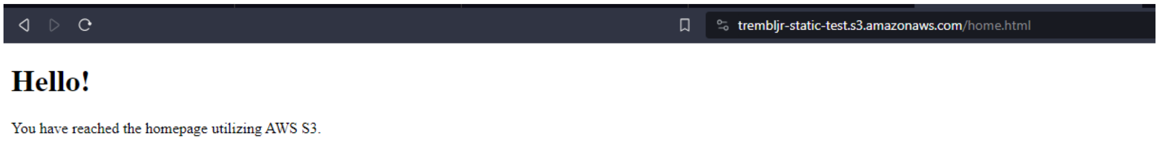
5.7. Security Testing

It is important to ensure the privacy and safety of user data and to maintain the accuracy and reliability of the predictions. In order to do that we will make sure that the dataset used for training and testing the model is anonymized and does not have any personally identifiable information. In order to do this we can implement data encryption mechanisms to protect the sensitive data during transmission and storage. We will also validate input data to prevent malicious input that could lead to incorrect predictions. We will lastly use authentication and authorization mechanisms to control user access to the model.

5.8. Results

We have completed basic tests to confirm that AWS can host an AI model. This uses the SageMaker service with its integrated Jupyter notebooks. Our test consisted of creating a notebook and writing some Python code to ensure that a basic, algorithmic machine learning model could be created and run on some data hosted in AWS. While our model will use neural networking, the test confirms that the foundational steps exist to create the model.

We have also tested hosting a front-end page utilizing a different AWS service called S3, a simple file storage system. This will help us keep our different components within the same overarching framework of AWS. As shown in figure 1, the link to the HTML page we created is publicly available, meaning that the service can be reached from anywhere once it is fully built out.


trembljr-static-test.s3.amazonaws.com/home.html

Hello!

You have reached the homepage utilizing AWS S3.

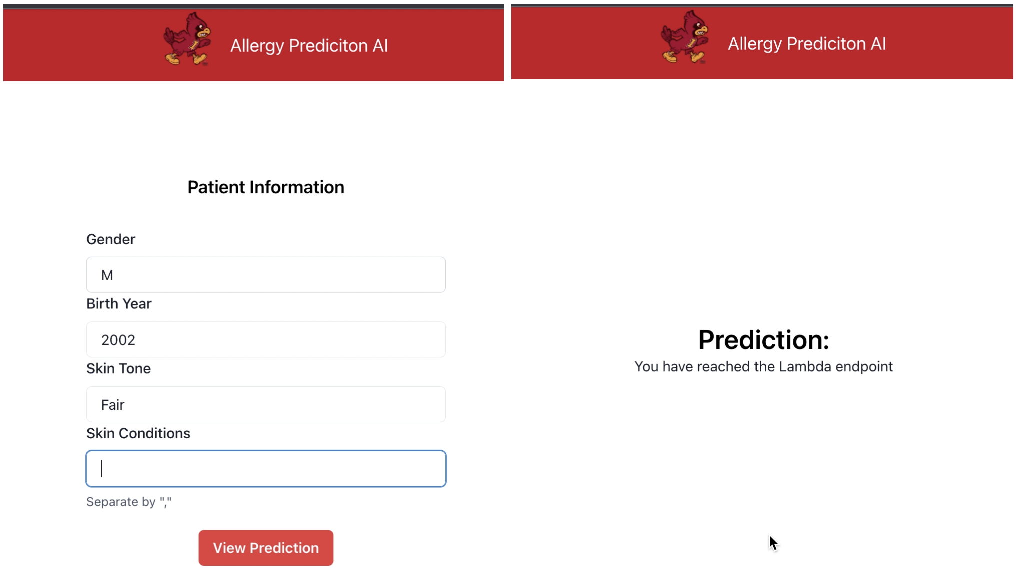
Figure 4: Result of basic front-end test using AWS to visually show an HTML page deployed to an AWS URL.

6. Implementation

The first section of our project we will build is our front-end. This section can be concurrently created with a simple backend, where we will have basic rules to detect allergic reactions instead of an AI for the early phase of the project. Once our project is capable of round-trip communication, the AI model can be trained and integrated into our project, and any additional time can be used to fine-tune the model and the overall system.

Front-end Client: Iteration 0

The simple React client takes in basic patient information and a button to send the form details to the backend AWS endpoint. Iteration 0 is intended as a demo prototype and future iterations will introduce additional and finalized input arguments required to feed into the AI model. The demo prediction message demonstrates end-to-end communication from the client to the server.


Allergy Prediciton AI

Patient Information

Gender: M

Birth Year: 2002

Skin Tone: Fair

Skin Conditions:

Separate by ","

Allergy Prediciton AI

Prediction:

You have reached the Lambda endpoint

Figure 5: Basic depiction of what the front-end may look like in the future with the connection to the backend demonstrated on the right.

7. Professionalism

7.1. Areas of Responsibility

Area of Responsibility	Definition	NSPE Canon	SE Code of Ethics	Difference
Work Competence	Perform work of high quality, integrity, timeliness, and professional competence.	Perform services only in areas of their competence; Avoid deceptive acts.	1.03 Ensure they are qualified, by an appropriate combination of education and experience, for any project on which they work or propose to work.	Both codes use similar wording to describe the proper conduct for work competence, but the NSPE actively states to avoid deception while the SE code only says to ensure appropriate qualification.
Financial Responsibility	Deliver products and services of realizable value and at reasonable costs.	Act for each employer or client as faithful agents or trustees.	1.07 Ensure realistic estimates of cost, scheduling, personnel, and outcome on any project on which they work or propose to work and provide a risk assessment of these estimates.	The SE code expands on what the NSPE Canon says, detailing the importance of accurate estimation with the projects.
Communication Honesty	Report work truthfully, without deception, and understandable to stakeholders.	Issue public statements only in an objective and truthful manner; Avoid deceptive acts.	2.06 Be fair and truthful in all statements, particularly public ones, concerning software or	Both the SE code and NSPE canon say the same thing, but use slightly different wording.

			related documents.	
Health, Safety, and Well-Being	Minimize risks to safety, health, and well-being of stakeholders.	Hold paramount the safety, health, and welfare of the public.	2.01 Disclose to appropriate persons or authorities any actual or potential danger to the user, a third party, or the environment, they reasonably believe to be associated with the software or related documents for which they are responsible, or merely know about.	The SE code goes into specifics for how to minimize risks. It gives a direct command to disclose any danger, and the NSPE canon says to uphold safety.
Property Ownership	Respect property, ideas, and information of clients and others.	Act for each employer or client as faithful agents or trustees.	5.05 Develop a fair agreement concerning ownership of any software, processes, research, writing, or other intellectual property to which an employee has contributed.	The SE Code for this section is clearer, as a specific action point is given (develop agreement), while the NSPE could be interpreted to be ambiguous.
Sustainability	Protect the environment and natural resources locally and globally.	N/A	2.02 Approve software only if they have a well-founded belief that it is safe, meets specifications, has passed appropriate tests, and does not diminish quality of life or	The NSPE canon is missing a way to address sustainability, while the SE code does have a guideline.

			harm the environment.	
Social Responsibility	Produce products and services that benefit society and communities.	Conduct themselves honorably, responsibly, ethically, and lawfully so as to enhance the honor, reputation, and usefulness of the profession.	6.06 Take responsibility for detecting, correcting, and reporting errors in software and associated documents on which they work.	The NSPE canon is a more complete representation of social responsibility. The SE code only has conduct when errors occur, but the NSPE canon can be followed at all times.

Table 3: Areas of responsibility in National Society of Professional Engineers (NSPE) compared to Software Engineering (SE) Code of Conduct.

7.2. Project Specific Responsibility Areas

Area of Responsibility	Project Context	Performance
Work Competence	Work competence should be expected in any project or task. This project is no exception, as creating a high-quality model requires a deep understanding of many concepts.	Medium, we are planning as best as we can and learning as we go, but it is to be expected we run into issues and change our plan accordingly to ensure we have the greatest chance to succeed
Financial Responsibility	While there is not much financial cost for the end user, there is still a cost to using a cloud provider like AWS. If not monitored carefully, the monetary cost of this service can have a large impact.	Medium, we have not had much to deal with for financial cost as of yet, but the areas we have considered financially are of adequate performance.
Communication Honesty	To ensure that all concerned parties understand AI and what it can do, we must communicate honestly and clearly.	High, our design is transparent and allows patients and professionals alike to see what data is being transmitted and what the results can be.

Health, Safety, Well-Being	Health, safety, and general well-being are all important to consider as they relate to the project, as an incorrectly functioning model that gives false medical information to users.	High. Our team has planned to output not only a prediction but also a transparent and impressive confidence rating that can be acted on, assuming the consumer is aware of and accepts the risks.
Property Ownership	Since the data used to train and test the model is personal and private, it places property ownership as something we need to be mindful of at all times.	High. Our team has already signed an NDA to receive the data we are using to train our model, and when questions arise over confidentiality or ethics, we are sure to ask them before acting.
Sustainability	While there are not many environmental concerns with the project, it is important to monitor resource usage as these can have an impact if not carefully observed.	Medium, similar to the financial responsibility, we have not had much to encounter with sustainability as of yet. Further research on the use of AWS vs Google Cloud environmental impacts once we know more about our final project's needs could determine which is best for our project.
Social Responsibility	This project, creating an artificial intelligence model to predict allergies, is closely aligned with the ethics of social responsibility. There is a direct benefit for many people, so it is important to ensure our product does not abuse that benefit.	High. Our project has a high potential to impact lives with low environmental, economic, and time costs. These benefits should be maximized while keeping the price affordable to help others feel better without excluding our product from those without health insurance or a high-paying job.

Table 4: Areas of responsibility in our project context and our current performance.

7.3. Most Applicable Responsibility Area

In the development of our AI model we decided that communication honesty, health, safety, well-being, and social responsibility are the most applicable responsibility areas. Communication honesty ensures that stakeholders, including users and healthcare professionals, receive accurate and transparent information about the model's capabilities and limitations. Prioritizing health, safety, and well-being is crucial to minimize risks associated with potential inaccuracies in predictions, safeguarding users from harm. Social Responsibility underscores the ethical duty to produce a model that benefits individuals. Integrating these principles ensures the ethical foundation of the AI model, promoting trustworthiness, user safety, and a positive societal impact.

8. Closing Material

8.1. Discussion

Since most of us were new to AWS and machine learning, we had to research different methods of implementing the model and necessary components in AWS. We used what we learned from our experiments and created a revised design of our project which will better suit our needs. One of the major changes we realized was necessary was the switch from Java to Python for our backend language. Python code is generally shorter than Java, resulting in faster compute time and lower costs of running. In addition, the model will be built using Python, so our team can develop in multiple areas without needing to learn or relearn a programming language first.

8.2. Conclusion

In addition to formulating a design and plan for implementation, we have tested the basic creation of each major component. Since we have tested the connection, our next step is making a new model that uses the real data to predict, along with remaking the major components for how we plan to use them. One constraint that we had while creating our design was the lack of data at the beginning of the semester. It was difficult to create a solid plan without knowing what we were working with. Fortunately, this has been resolved, and we were able to update our plan and design accordingly.

8.3. References

[1] *Medical device software — Software life cycle processes*, IEC Standard 62304, 2006.

[2] *Health informatics–Device interoperability–Part 20601: Personal health device communication–Application profile –Optimized exchange protocol*, IEEE Standard 11073, EMB/11073 - IEEE 11073 Standards Committee, 2022.

[3] *IEEE Recommended Practice for the Quality Management of Datasets for Medical Artificial Intelligence*, IEEE Standard 2801, EMB/Stds Com - Standards Committee, 2022.

[4] M. van Breugel et al. "Current state and prospects of artificial intelligence in allergy". *Allergy*. 2023; 78: 2623-2643. https://doi.org/10.1111/all.15849.

8.4. Appendices

8.4.1. Team Contract

Team Members:

1) Joseph Trembley   2) Noah Ross   3) Xerxes Tarman   4) Ella Godfrey   5) Alex Ong

Team Procedures

We will meet every Wednesday from 4:30-5:30 pm virtually to discuss our current progress and talk through any issues we have.

We will communicate using a text group chat and Discord for meetings. We will use email to contact the client, and we use Webex for meetings with the client.

If a conflict occurs, we have 5 people, so there will always be a leading majority. Resolutions will be attempted before a vote, but for time's sake, majority votes may be necessary.

We will use a shared Google doc to keep track of the minutes. Ella will keep track of the minutes. For bigger milestones we will also add them to the git so they can be tracked in multiple ways.

Participation Expectations

If a team member can not make a scheduled meeting, they must inform the team 24 hours in advance.

If a team member does not hit a deadline, we will have a team conversation to understand the circumstances, but if the situation becomes habitual, further action from an advisor will be requested.

At least once a day, each group member should check our primary means of communication (texting, discord, email).

Team members should be willing to compromise on both tasks and decisions between what has to be done and what they would like to do. We are expected to take pride in our work.

Leadership

For current leadership roles, Joseph has taken on the team lead role. This role organizes client meetings and sends any emails that need to be sent to the client. Noah's role will be any team organization. This will include setting team meetings and anything that has to do with the team. Ella has taken the role of taking minutes for the team. As the project progresses the leadership roles will be updated and added to. This is a beginning list of roles we have so far, but it will continue to grow as we continue the project.

To support and guide all team members effectively, we must maintain clear communication, have individualized goals, and have regular feedback. If team members aren't cooperating, we will address it as a group and discuss it with that teammate to aid in overcoming any obstacles they may be facing. We will promptly address any non-cooperation to keep our project flowing.

Many of the technologies we will be using (Google Docs, GitLab, Discord) all have a way of showing who has done which parts/has been in frequent contact with the group. If there appears to be a discrepancy between what we have and what someone says they did, we can check these logs. For any stories/projects that require multiple people working at the same time, we will have those team members say what they believe to be a fair division of work.

Collaboration and Inclusion

Joseph has 6 summers of experience interning in a technology department for a large-scale company. In that time, he has had the opportunity to truly learn Java, Spring Boot, and databases. He has also achieved the AWS Cloud Practitioner certification. Ella has experience with front-end development working for a company focused on web development. Noah has 2 summers of experience developing User Interfaces on construction/agriculture products. His strengths lay in human interaction and defining the needs and wants of customers. Xerxes has two summer internships working as an embedded software developer. He has experience collaborating and working on an agile team. Alex has two summer internships, and one has continued part-time during the school year totaling nearly 1.5 years. He has worked on industry front-end and backend projects and designed personal websites and e-commerce sites, giving a range of full-stack development skills.

After reporting on our current progress, our meetings will go over what needs to be done in the future. We will discuss future steps, and if team members have ideas, they can step in and say how they believe the task should be done. This way, we can ensure everyone has the opportunity to contribute to planning the stories.

The beginnings of our meetings are dedicated to discussing our current progress in our work and anything that is preventing positive progress from being made. It should be at this time when a team member mentions whether an outside factor is influencing their ability to work. If the issue is more urgent and needs to be resolved sooner, the team member will send a message explaining the situation.

Goal-Setting, Planning, and Execution

We wish to learn more about the machine learning process, understand our options with this technology, and plan on how to integrate it into our project for next semester. Learning best practices for delivering results to a real customer will translate well to industry work.

Our main strategy for assigning work is to define the problems and then assign tasks based on the skill level and comfort level of the team members.

We plan on working in sprints and setting time-sensitive goals for our project.

Consequences for Not Adhering to Team Contract

If there are infractions to the team contract we will discuss them as a team to try to fix any misinterpretation. If the same person keeps having multiple infractions we might have to talk with the professor, who can be more strict with this. This could also result in a bad team evaluation.

We will first seek a conversation with the perpetrator, but if no progress is made, we will seek guidance from advisors and the TA.

****

a) I participated in formulating the standards, roles, and procedures as stated in this contract.

b) I understand that I am obligated to abide by these terms and conditions.

c) I understand that if I do not abide by these terms and conditions, I will suffer the

consequences as stated in this contract.

1) Ella Godfrey ____ DATE 9/6/23______________

2) Xerxes Tarman_______________________________________DATE 9/6/23_______________

3) Joseph Trembley_____________________________________ DATE 9/6/23_______________

4) Noah Ross____ DATE 9/6/23_______________

5) Alex Ong____ DATE 9/6/23_______________