I. Introduction / Overview.

Currently, the world is increasingly developing machines and applications that have met most of the needs of people, besides, new diseases are also increasing. We can recognize health care as a very important need. Especially now that people often ignore small signs of pain and often do not go to the hospital for many reasons such as work, money. But they can be signs of serious illness.

A health care application for people quickly, economically, anytime, anywhere is a necessity. In addition, the application will help users contact the surrounding hospitals when needed. Besides, the application will also generate additional income for doctors when they complete the tasks in the application.

One of the problems that I will have to deal with is the initial data pre-construction and processing. To be able to build a patient diagnosis model, the first thing I need to do is build a dataset with some common diseases that people often encounter. Data labeling is also quite difficult because to label a disease you need knowledge about that disease. After generating and pre-processing the data, the next step is to build a model with high accuracy. Since the model doesn't need to run in real time and the input is just an image, I'm more concerned with the accuracy than the weight of the model. The difficulty here is that I need to build a CNN model to classify the disease and locate it, and then pass the image through the compatible model to get the most accurate disease shape.

To solve the problem of data set labeling. First, I will try to build a model to predict the skin diseases that people commonly suffer from. This makes it easier for users to access and use the application, then when the user uses the application and allows to provide some disease images, qualified doctors will use them and label them. is their task on the app, every time they complete the task they will get a worthy reward. As for disease prediction, this part I am only interested in accuracy because I do not need to predict the disease in real time. First, I will feed the dataset through a CNN model for disease classification and location identification. For this section, I will study recent and highly accurate articles such as yolov5, R-CNN, FasterR-CNN, and their improved versions. Then the data will be segmented to be able to get the shape of the disease. In this part, I will use separate models for each disease

and research models such as Unet, Deeplapv3 to increase the accuracy to the highest.

Key phrases : Tensorflow, Pytorch, Front-end, Back end.

II. Aim.

I created this skin disease diagnostic application with the aim of being able to help users take care of their own health quickly and easily, protect users from emergencies of dangerous diseases, provide effective methods for users when sick. At the same time, build a database from user-approved images, provide it for research and training purposes, to create machines that really look like a good doctor that enhances diagnosis .

III. Objectives.

1. Analyze user requirements. Activities :

- Investigate some people who don't often go to the doctor, the reason they don't go to the hospital and their thoughts on their applications and aspirations.

- Interview patients with skin diseases and their wishes about the application. ask users about their common skin diseases.

- Analyze user requirements.

Deliverable:

- List and analyze the features most desired by users.

2. Data collection Activities :

- Perform research on classification models, detect as yolov5 and previous diagnostic models such as deeplapv3.

- Research, select methods to choose the best model.

- Collection and processing of common skin disease imaging data.

Deliverable:

- Report on collected data

- Create the best end to end model for disease diagnosis.

3. Technology applied in the application. Activities :

- Front-end technology used : android (java), ReactJs

- Back-end technology used: python, nodeJs

- Model Ai will be create by: python, C++

- Use database: Mysql, Nosql

- API used in the application: restful

Deliverable:

- Compare and choose the best technology.

4. System architecture design Activities :

- Database design for "my doctor" app, The database will include information about the user, the doctor and information about the photos that the patient provides to the app

- Model design for "my doctor" app. Design a disease prediction model, focusing on skin diseases

- Functional design for "my doctor" app, Functions aimed at healthcare goals for users

- Interface design for "my doctor" app

- Structural design for "my doctor" app

Deliverable:

- Report the parts of database design, functionality, user interface, system structure.

5. Implement the "my doctor" app. Activities :

- Implement model skin disease prediction for "my doctor" app

- Implement database including information about doctors, users for "my doctor" app

- Implement back-end for "my doctor" app

- Implement front-end for "my doctor" app

- Implement health care functions for "my doctor" app

Deliverable:

- Complete all health care functions of the application.

6. Test "my doctor" app. Activities :

- Create accuracy test for model disease location, disease classification, disease segmentation

- Test cases for the app's healthcare functions.

- User interface testing for “my doctor” app.

- Test API for “my doctor” app.

Deliverable:

- The report includes the results of all tests.

7. Roll out application.

Activities :

- Complete detailed user guide.

- Complete technical documentation.

- System maintenance.

- Complete the project on time.

Deliverable:

- Detailed reports on progress and completed works.

IV. Legal, Social, Ethical and Professional.

1. legal.

About the legality of the application. All personal information when users register to use the application, disease status, skin disease photos, medical records will be guaranteed not to be shared with anyone or any organization. All such information is only allowed to serve the

needs of individual users. For information that may be in public form such as doctor's phone number, doctor's certificate, the doctor's information will only be used when the user registers and allows the application to use that information. . Information shared from users is only for the functionality of the application and not for commercial purposes**.**

2. Social.

About the app's social. The application helps create a generation of users who care about their health habits, by checking for skin diseases and diagnosing what they are and whether they are signs of dangerous diseases. No matter how difficult it is in terms of money or time, they can take care of their own health all the time. Create a habit for users to always check their own health with AI diagnostics and doctors are always available to advise.

3. Ethical.

About the ethics of the application. Ensure the user's skin disease image information is kept confidential in the best way. Ensure that a team of qualified doctors will effectively advise users, all information about prices after each consultation and medical examination is made public. All costs will be guaranteed to be the most reasonable for users. Any questions and feedback from users will be resolved by the manager as quickly as possible.

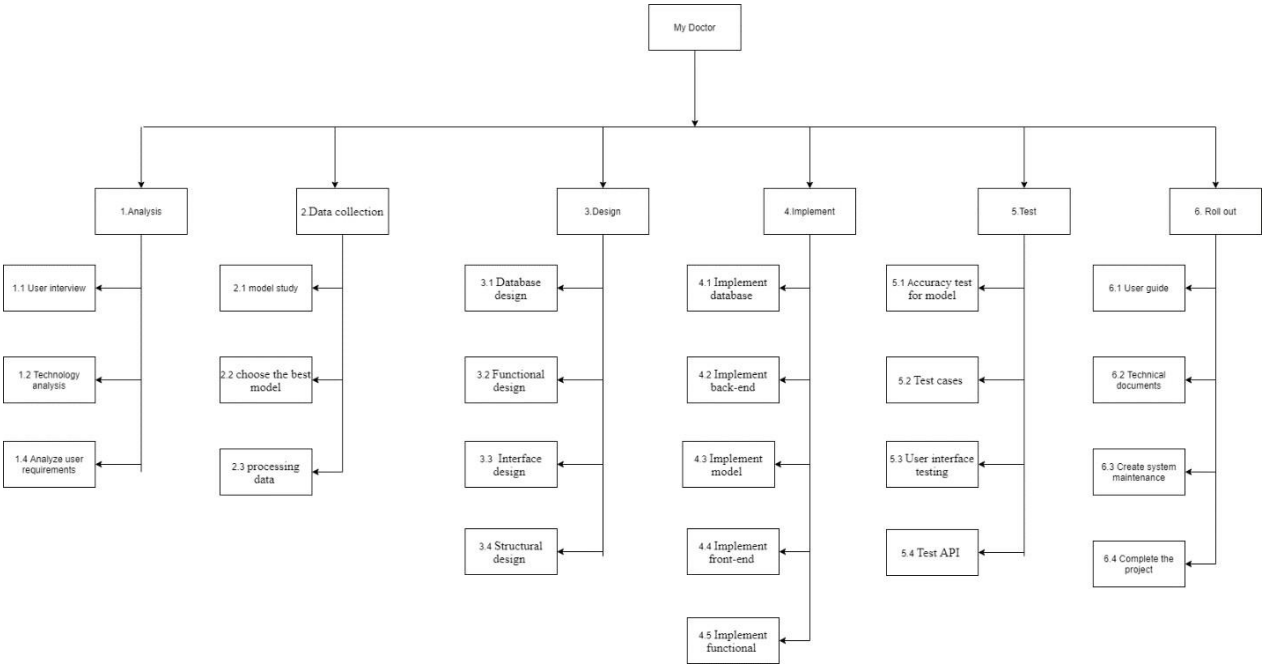
4. Professional.

Ensure the safety of users' personal data about skin disease photos, medical records, other personal information in the best way. Any feedback and comments from users about the diagnostic functions in addition, the doctor will be answered quickly. New functions and versions when being updated are all for the purpose of taking care of the health of the user, and will notify the user in detail and quickly. Application errors will be fixed as quickly as possible. The skin disease diagnostic model will be continuously learning data so that the user can get the

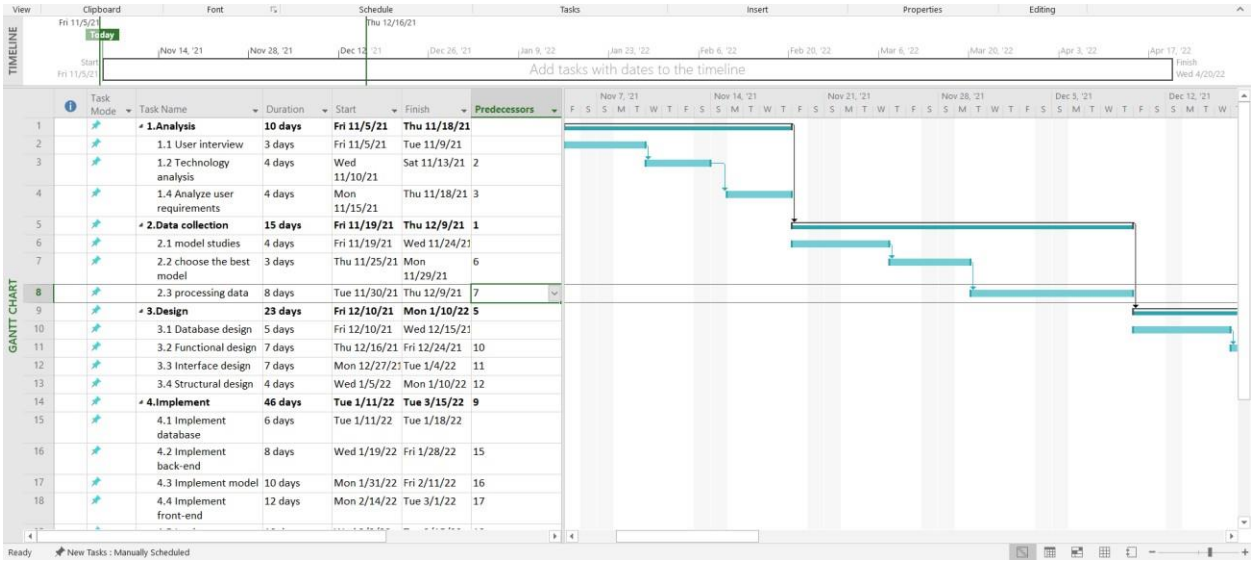
best results from the prediction functions. Any errors from doctors, apps, and managers will be publicly resolved.

V. Plan.

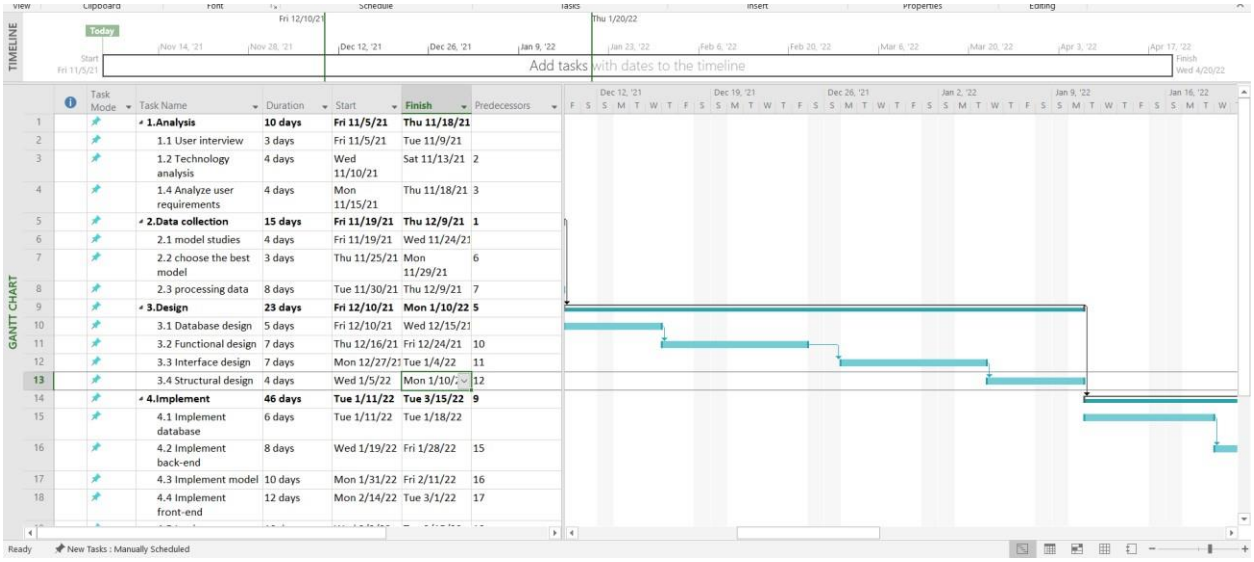
1.WBS.



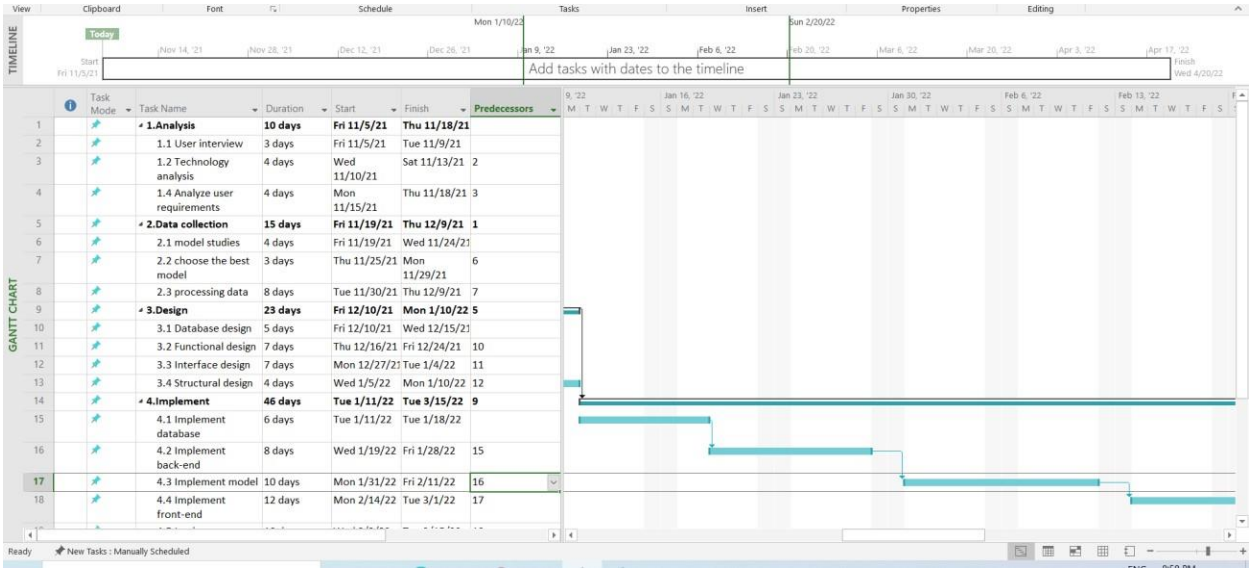
2. Gantt Chart.



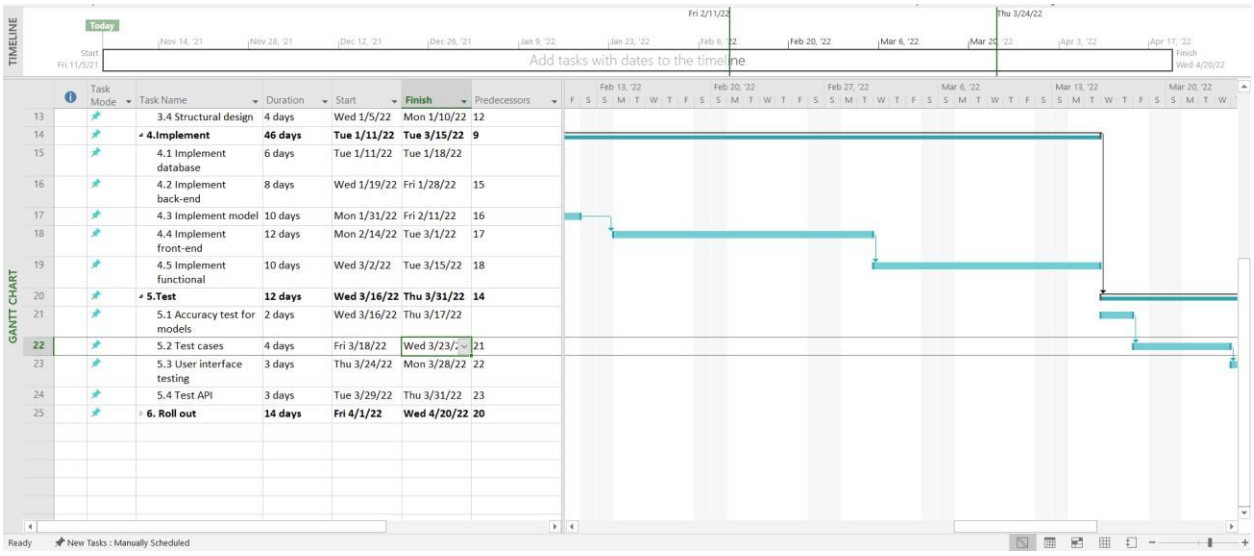
	Task Mode	Task Name	Duration	Start	Finish	Predecessors
1		**1.Analysis**	**10 days**	**Fri 11/5/21**	**Thu 11/18/21**	
2		1.1 User interview	3 days	Fri 11/5/21	Tue 11/9/21	
3		1.2 Technology analysis	4 days	Wed 11/10/21	Sat 11/13/21	2
4		1.4 Analyze user requirements	4 days	Mon 11/15/21	Thu 11/18/21	3
5		**2.Data collection**	**15 days**	**Fri 11/19/21**	**Thu 12/9/21**	**1**
6		2.1 model studies	4 days	Fri 11/19/21	Wed 11/24/21	
7		2.2 choose the best model	3 days	Thu 11/25/21	Mon 11/29/21	6
8		2.3 processing data	8 days	Tue 11/30/21	Thu 12/9/21	7
9		**3.Design**	**23 days**	**Fri 12/10/21**	**Mon 1/10/22**	**5**
10		3.1 Database design	5 days	Fri 12/10/21	Wed 12/15/21	
11		3.2 Functional design	7 days	Thu 12/16/21	Fri 12/24/21	10
12		3.3 Interface design	7 days	Mon 12/27/21	Tue 1/4/22	11
13		3.4 Structural design	4 days	Wed 1/5/22	Mon 1/10/22	12
14		**4.Implement**	**46 days**	**Tue 1/11/22**	**Tue 3/15/22**	**9**
15		4.1 Implement database	6 days	Tue 1/11/22	Tue 1/18/22	
16		4.2 Implement back-end	8 days	Wed 1/19/22	Fri 1/28/22	15
17		4.3 Implement model	10 days	Mon 1/31/22	Fri 2/11/22	16
18		4.4 Implement front-end	12 days	Mon 2/14/22	Tue 3/1/22	17



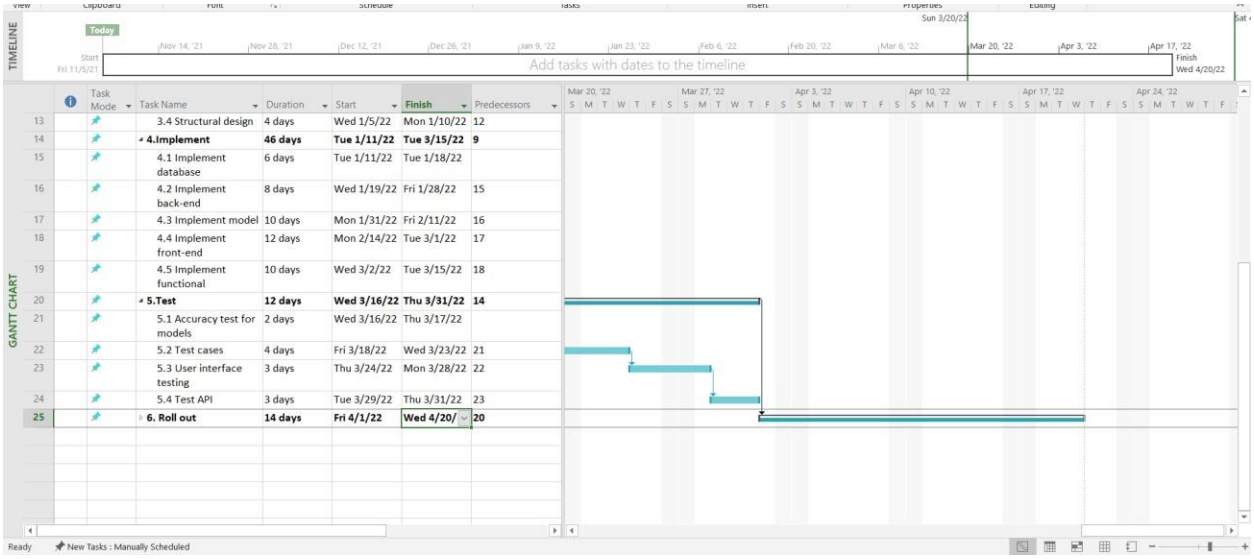
	Task Mode	Task Name	Duration	Start	Finish	Predecessors
1		**1.Analysis**	**10 days**	**Fri 11/5/21**	**Thu 11/18/21**	
2		1.1 User interview	3 days	Fri 11/5/21	Tue 11/9/21	
3		1.2 Technology analysis	4 days	Wed 11/10/21	Sat 11/13/21	2
4		1.4 Analyze user requirements	4 days	Mon 11/15/21	Thu 11/18/21	3
5		**2.Data collection**	**15 days**	**Fri 11/19/21**	**Thu 12/9/21**	**1**
6		2.1 model studies	4 days	Fri 11/19/21	Wed 11/24/21	
7		2.2 choose the best model	3 days	Thu 11/25/21	Mon 11/29/21	6
8		2.3 processing data	8 days	Tue 11/30/21	Thu 12/9/21	7
9		**3.Design**	**23 days**	**Fri 12/10/21**	**Mon 1/10/22**	**5**
10		3.1 Database design	5 days	Fri 12/10/21	Wed 12/15/21	
11		3.2 Functional design	7 days	Thu 12/16/21	Fri 12/24/21	10
12		3.3 Interface design	7 days	Mon 12/27/21	Tue 1/4/22	11
13		3.4 Structural design	4 days	Wed 1/5/22	Mon 1/10/22	12
14		**4.Implement**	**46 days**	**Tue 1/11/22**	**Tue 3/15/22**	**9**
15		4.1 Implement database	6 days	Tue 1/11/22	Tue 1/18/22	
16		4.2 Implement back-end	8 days	Wed 1/19/22	Fri 1/28/22	15
17		4.3 Implement model	10 days	Mon 1/31/22	Fri 2/11/22	16
18		4.4 Implement front-end	12 days	Mon 2/14/22	Tue 3/1/22	17



	Task Mode	Task Name	Duration	Start	Finish	Predecessors
1		**1.Analysis**	**10 days**	**Fri 11/5/21**	**Thu 11/18/21**	
2		1.1 User interview	3 days	Fri 11/5/21	Tue 11/9/21	
3		1.2 Technology analysis	4 days	Wed 11/10/21	Sat 11/13/21	2
4		1.4 Analyze user requirements	4 days	Mon 11/15/21	Thu 11/18/21	3
5		**2.Data collection**	**15 days**	**Fri 11/19/21**	**Thu 12/9/21**	**1**
6		2.1 model studies	4 days	Fri 11/19/21	Wed 11/24/21	
7		2.2 choose the best model	3 days	Thu 11/25/21	Mon 11/29/21	6
8		2.3 processing data	8 days	Tue 11/30/21	Thu 12/9/21	7
9		**3.Design**	**23 days**	**Fri 12/10/21**	**Mon 1/10/22**	**5**
10		3.1 Database design	5 days	Fri 12/10/21	Wed 12/15/21	
11		3.2 Functional design	7 days	Thu 12/16/21	Fri 12/24/21	10
12		3.3 Interface design	7 days	Mon 12/27/21	Tue 1/4/22	11
13		3.4 Structural design	4 days	Wed 1/5/22	Mon 1/10/22	12
14		**4.Implement**	**46 days**	**Tue 1/11/22**	**Tue 3/15/22**	**9**
15		4.1 Implement database	6 days	Tue 1/11/22	Tue 1/18/22	
16		4.2 Implement back-end	8 days	Wed 1/19/22	Fri 1/28/22	15
17		4.3 Implement model	10 days	Mon 1/31/22	Fri 2/11/22	16
18		4.4 Implement front-end	12 days	Mon 2/14/22	Tue 3/1/22	17



	Task Mode	Task Name	Duration	Start	Finish	Predecessors
13		3.4 Structural design	4 days	Wed 1/5/22	Mon 1/10/22	12
14		**4.Implement**	**46 days**	**Tue 1/11/22**	**Tue 3/15/22**	**9**
15		4.1 Implement database	6 days	Tue 1/11/22	Tue 1/18/22	
16		4.2 Implement back-end	8 days	Wed 1/19/22	Fri 1/28/22	15
17		4.3 Implement model	10 days	Mon 1/31/22	Fri 2/11/22	16
18		4.4 Implement front-end	12 days	Mon 2/14/22	Tue 3/1/22	17
19		4.5 Implement functional	10 days	Wed 3/2/22	Tue 3/15/22	18
20		**5.Test**	**12 days**	**Wed 3/16/22**	**Thu 3/31/22**	**14**
21		5.1 Accuracy test for models	2 days	Wed 3/16/22	Thu 3/17/22	
22		5.2 Test cases	4 days	Fri 3/18/22	Wed 3/23/22	21
23		5.3 User interface testing	3 days	Thu 3/24/22	Mon 3/28/22	22
24		5.4 Test API	3 days	Tue 3/29/22	Thu 3/31/22	23
25		**6. Roll out**	**14 days**	**Fri 4/1/22**	**Wed 4/20/22**	**20**



VI. Initial References.

Scacca, S., 2019. *telerik.* [Online] Available at: https://www.telerik.com [Accessed 5 November 2021].

Varshneya, R., 2019. *entrepreneur.* [Online]
Available at: https://www.entrepreneur.com/article/231145 [Accessed 5 November 2021].