Hey Nurse!

an Intelligent Nurse Call System

NUS Master of Technology
Intelligent Reasoning System (IRS)

Brought to you by 'Team FARY'

Yap Jing Yang A0229976H
Francis Tan Wee Meng A0229963N
Ramya Shanmugam A0229958H
Wang Dongchen A0083129E

2-May-2021

Table of Content

1 Executive Summary

Singapore, a small nation of over 5.5 million, faces a shortage of nursing professionals due to the population growth, rapid aging, increasing burden from chronic diseases, and less people taking up the profession. For those who are already in the profession, the working conditions, fatigue, distraction and interruption while performing their work can contribute to the increase in stress level, and affects the cognitive, and decision-making abilities to the point that it begins to threaten the safety of the patients. As such, any effort to reduce the stress and improve the working conditions for the nursing professional is always going to be viewed and received in the positive light.

Our Hey Nurse! an intelligent Nurse Call system has been conceived and designed for nurses to provide a high standard and quality of care at a much lower level of stress. A voice activated listening device build on Mycroft open-source technology is used to capture the audio input from the patient. One of the highlights of Hey Nurse! is the machine learning capability that allows for automated prioritization of the patient's request in a hospital ward. By presenting the patients' requests on a minimalistic but yet user-friendly nurse-dashboard, Hey Nurse! is meant to assist the nurses to plan and review their actions before attending to the patients. At the same time, nurses with the right level of experience can be assigned to attend to the various requests. The collected data accumulated over time by the Hey Nurse! system can assist the hospital to plan their nursing resources more effectively and to optimise the composition of their nursing teams.

This first iteration of Hey Nurse! is a prototype that serves as a proof-of-concept and future improvements include adding functionalities such as extracting emotion information out of the patient audio input, building two-ways communication between nurses and patients, and many more. Refer to Section 6.2 for more details.

2 Business Case

2.1 Problem Statement

Hospitalized patients are very vulnerable, and they require a lot of attention and care from the nurses. In the current hospital healthcare system, the nature of the request from the patients is not known until the nurse attends to the patient.

Sometimes the request can be as trivial as bringing a glass of water to the patient but there will be wasted time as the nurse will need to walk over to the patient first to find out about the request.

Because the manpower pool is so limited, the nurses will be overwhelmed in the vast sea of different requests in the huge patient population. Especially in the current COVID world where there are additional safety measures imposed, the nurse manpower pool will even be more stretched.

2.2 Proposed Solution

2.2.1 Project Objectives

Our team has brainstormed and came up with a proposed AI solution to help address the issues.
The objective of our project is to design and integrate an intelligent system, which will help nurses to respond to patients in their ward more efficiently by providing recommendations prior to nurses picking up the requests from their workstation.

Our solution proposed the following objectives:

1. **Prioritize & Streamline Workflow**. Help the nurses to prioritize and streamline their workflow, reducing the gap time in responding to patients' requests.

2. **Reduced Waiting Times**. Optimize patient waiting time for nurses in a hospital ward.

3. **Make Better Decisions**. Help nurses make better decisions based on prior information fed from our intelligent system so as to prepare themselves better before going to the ward.

4. **Constant Refinement**. With the build-up of growing dataset, our recommendation system can be constantly refined to produce more accurate classification of the in-coming requests.

2.2.2 Market Research

AI in healthcare is an emerging market and is expected to reach $6.6 billion by 2021. The top three AI applications consist of virtual nursing assistances ($20 billion) and administrative workflow assistance ($18 billion)[1].

In the current hospital practice, inpatients' requests are signaled to nurses by call bell in Singapore. In other countries, there are smart call nurse systems available in the market. However, patients have to use a mobile/tablet interface to send in their requests. The system can be too complex and hard to use for all patients under different kinds of medical conditions.

[1] Reference: https://www.accenture.com/sg-en/insight-artificial-intelligence-healthcare

Voice calling system is also available in the market, however these systems only servers a simple two-way communication channel. It did not process the calls from the patients further.

Our product targets at this $38 billion potential market through being an intelligent nurse assistant and reducing unnecessary administrative work.

The table below pairs up our proposed product versus the available nurse call systems in the market.

	User Friendliness	Voice Activation	Request Classification with Built-in Intelligence	Trainability
Hey Nurse!	Yes	Yes	Yes	System can be trained with more data
Best Group	Complicated interface, needs IT literacy	No voice activation	Predefined priority provided for patients to choose from	No
BEC	Same caller system as in current hospital	No voice activation	No	No

2.3 Project Team

Full Name	Work Items (Who Did What)
Yap Jing Yang	• Project idea generation • Implementation of 'Nurse Dashboard' frontend interface using Angular • Debug & troubleshooting • Business video • Project report writing
Ramya Shanmugam	• Technical Lead • Project idea generation • Basic App setup • Mycroft.AI – Implementation to capture user voice input and convert to text file. • Implementation of 'Admin Dashboard' frontend interface using Angular • SendGrid integration for sending email.

	• Implementation of the Query, Insert & Update APIs (User and Session). • Debug & troubleshooting • Technical video • Project report writing
Francis	• Project idea generation • Implementation of the Query, Insert & Update APIs (Requests) • Deployment of MongoDB database • Debug & troubleshooting • Technical video • Project report writing
Wang Dongchen	• Project idea generation, market research • Text classification: extract relevant text from the input and classify them according to importance level • Generate knowledge base for text classification training • Debug & troubleshooting • Business video • Project report writing

3 Product Design

3.1 Main Features

As stated in section 2.1, the main goal of this group project is to integrate an intelligent **Nurse Call** system using various open-source libraries and frameworks available online. The system takes in voice input from the patients and process it into meaningful information at the **Nurse Dashboard**.

Nurses are well-prepared before attending to patient. Help nurses to prepare in advance before attending to patients based on message contents.

Recommends case severity to Nurses. The system will recommend case severity to nurses so that nurses may attend to more urgent cases first.

More accurate and better recommendations. When more data is collected, our model will be refined to produce more accurate and better recommendations.

3.2 Process Flow

Before we go into the technical implementation of the project, designing a flowchart is a good way for us to understand the overall high-level structure of the intelligent system, as shown in the diagram below.

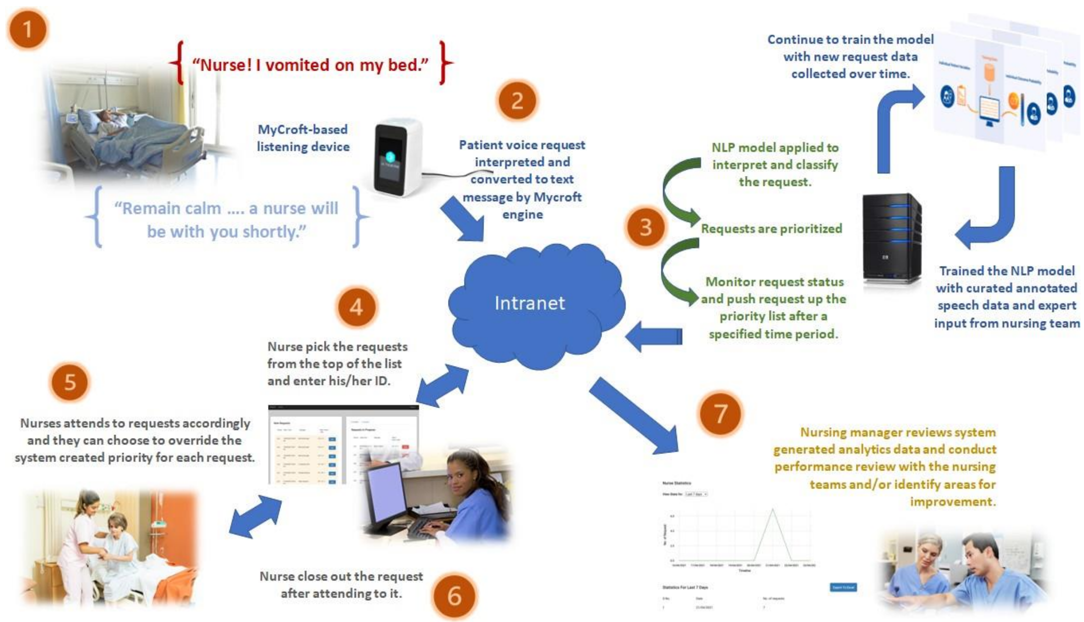


Figure 1: General Workflow of Hey Nurse!

Hey Nurse! will pick up the voice request from the patient and perform processing to convert it into a text input. In the next system, our algorithm will process the text using Natural Language Processing (NLP) and recommend a priority level for the given request. The processed message will then appear on the screen of the Nurse Dashboard and the nurse can view and action on the request. All the data will be stored into the database to facilitate refinement of the recommender model.

3.3 System Highlights

3.3.1 Voice Activation

Our system is activated by voice. Patients can send in their request by simply calling nurse in their rooms. The voice message will be processed to text message and sent for further processing.

Talking is the simplest yet the most intuitive way of asking for help and it is suitable for all patients' groups. Their requests will be registered once their voice is picked up by our MyCroft.AI receiver. Patients do have to reach for call bells. The issues caused by patients cannot reach call bells will be minimized.

3.3.2 Patients’ Request Processing and Reporting

The patients’ requests will be sent to the request classification module and classified according to the emergency/seriousness of the request. The classified request will then be seen in the Nurse Dashboard. Requests of higher priority will appear at the top of the list and not attended requests will flash yellow color after a certain time limit.

This feature helps nurses to get ready the materials needed by the patients before attending to the patients. It helps the nurse to reduce the time spent on travelling to and forth between the ward and the nursing station. Nurses can also prioritize and streamline their workflow. Patients who have similar requests could be attended together by the same specialist. Work efficiency will be greatly increased.

3.3.3 Data Collection and Re-Learning

Feedbacks on the request classification priority will be collected from the nurse explicitly through survey and implicitly through nurse selection of task. If the nurse always chooses a certain Class 2 request instead of a Class 3 request, the classification of the Class 2 request will be reviewed. Relearning of the request classification module will take place periodically to improve system accuracy.

4 System Modelling

4.1 Technical Flow

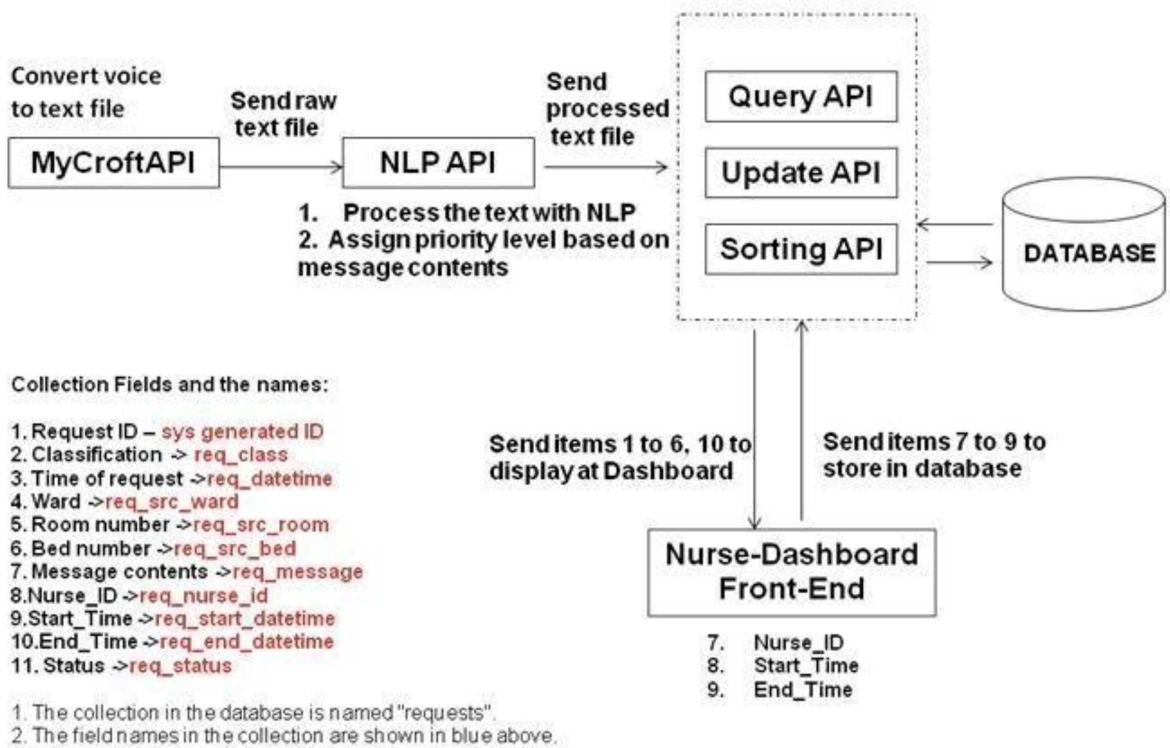


Figure 2: Technical workflow of our system.

The diagram above is showing the technical workflow of our system. The voice of the patient will trigger our receiver, Mycroft API which will convert the voice call into the text file and then pass the file to our NLP processing Logic which will perform the text processing and assign it a priority level. The processed message will then flow to our backend logic which will then push it into the database.

The backend logic will perform the interactions with the database based on the requirements from the Nurse-Dashboard which is our front-end that our user interacts with. The detailed explanations of the various subsystems are given in chapter 5.

5 System Development & Implementation

5.1 User Interface

5.1.1 Mycroft.AI (User input): https://github.com/ramya0311/nurse-assitant-skill

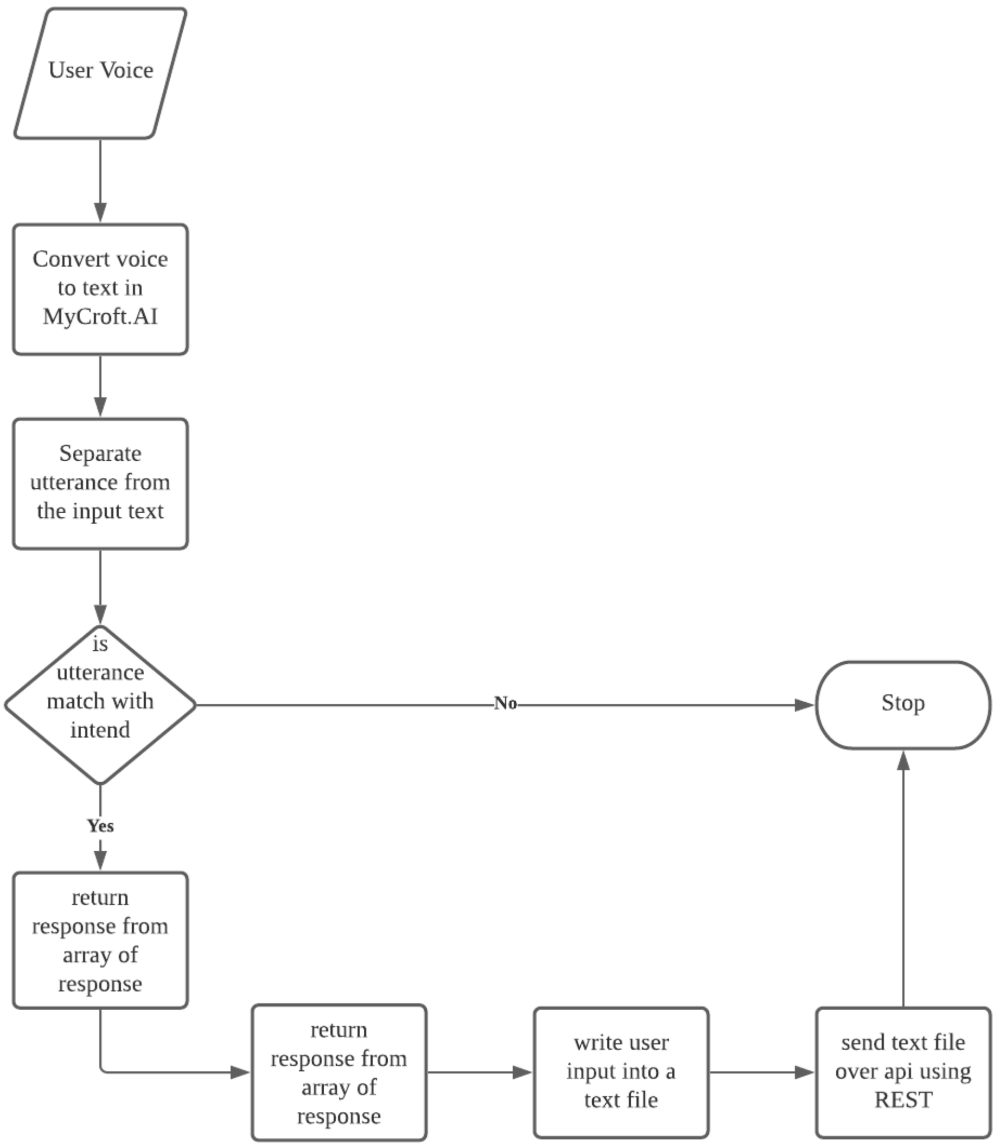


Figure 3: Technical workflow of MyCroft.AI system.

Mycroft.AI is an open source voice assistant (Cognitive System), which runs on a desktop, laptop or Raspberry Pi. A Mycroft Skill is a voice application that provide different functionalities for users. Mycroft allows us to create new skills using python.

Mycroft is activated when the wake word is spoken. It then listens to any query or command or phrase that the user is speaking, this phrase is called an ‘utterance’.

Once an utterance is captured by Mycroft, it matches it against all the predefined intents that’s available in the skills installed in it by identifying the ‘intent’ and gets the Skill that the intent it belongs to.

If a skill is identified, a random response from the predefined set of ‘dialog’ is chosen by Mycroft and is converted to speech. This is the response that the user will hear once an utterance is made. In the background Mycroft will start to process the user’s speech into text and continue with executing the logic that the skill is written for.

In our ‘Hey Nurse!’ application, we are using Mycroft to gather user input such as a patient’s request when the wake word ‘Christopher’ is called, followed by a request, (e.g.) ‘I fell down’, a response from the set of dialogs is chosen and spoken by Mycroft such as ‘Someone will be with you shortly to assist you’. In the background it would start to convert the voice input into a text file, which in turn is sent to the python API created in ‘Hey Nurse’ code base, where it is processed and later saved as a collection into the MongoDB database.

5.1.2 Nurse Station Panel

The frontend was developed using Angular framework. Angular is a platform and framework for building single-page client applications using HTML and TypeScript.

The panel will display the list of new requests and in-progress requests for the nurses to view and attend to. The ‘New Requests’ table displays the list of new requests and the ‘Requests in Progress’ table displays the current requests that have been taken up by the nurse but not yet complete.

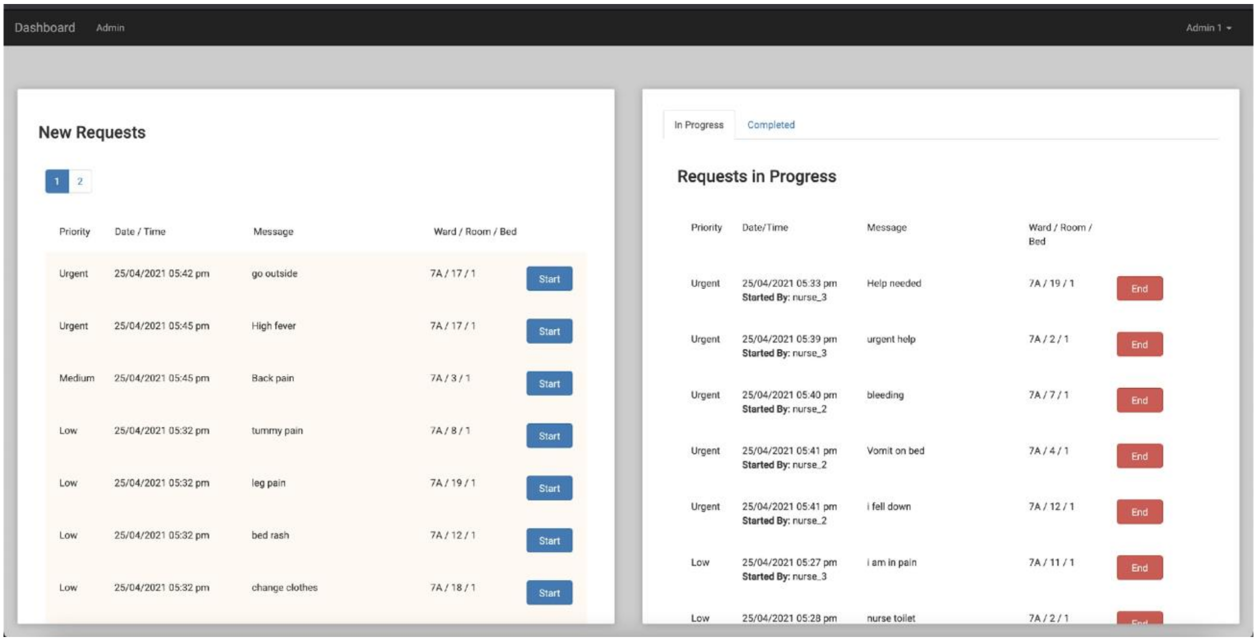


Figure 4: The Nurse Dashboard user interface

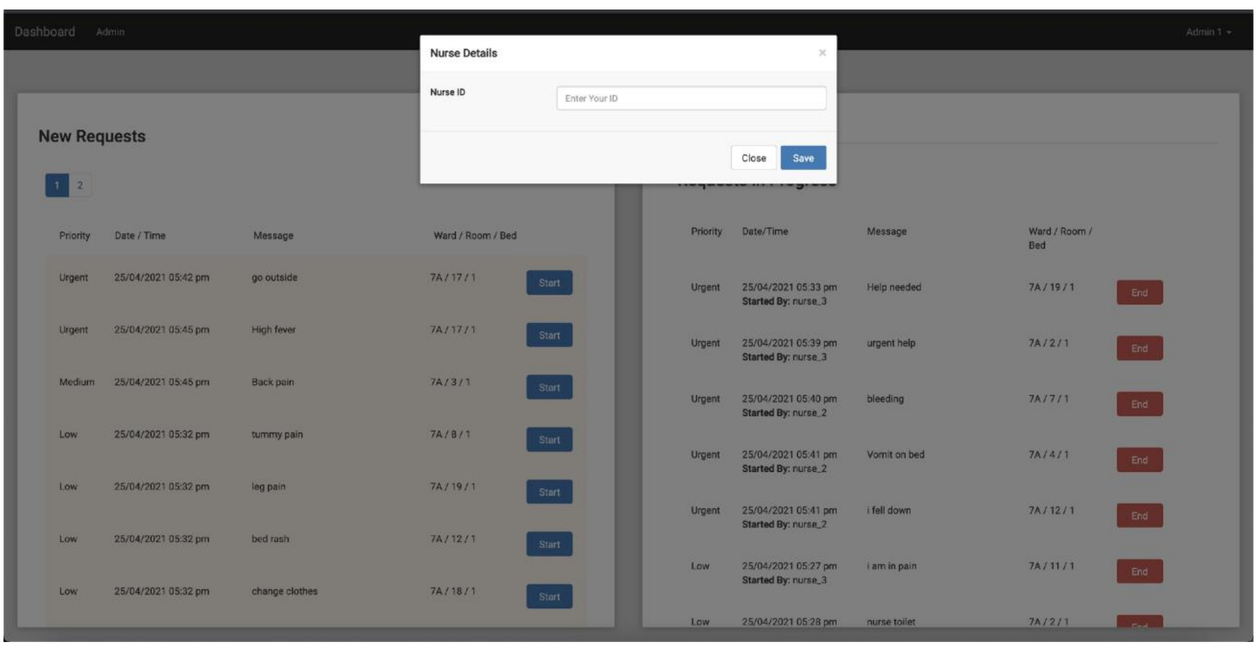
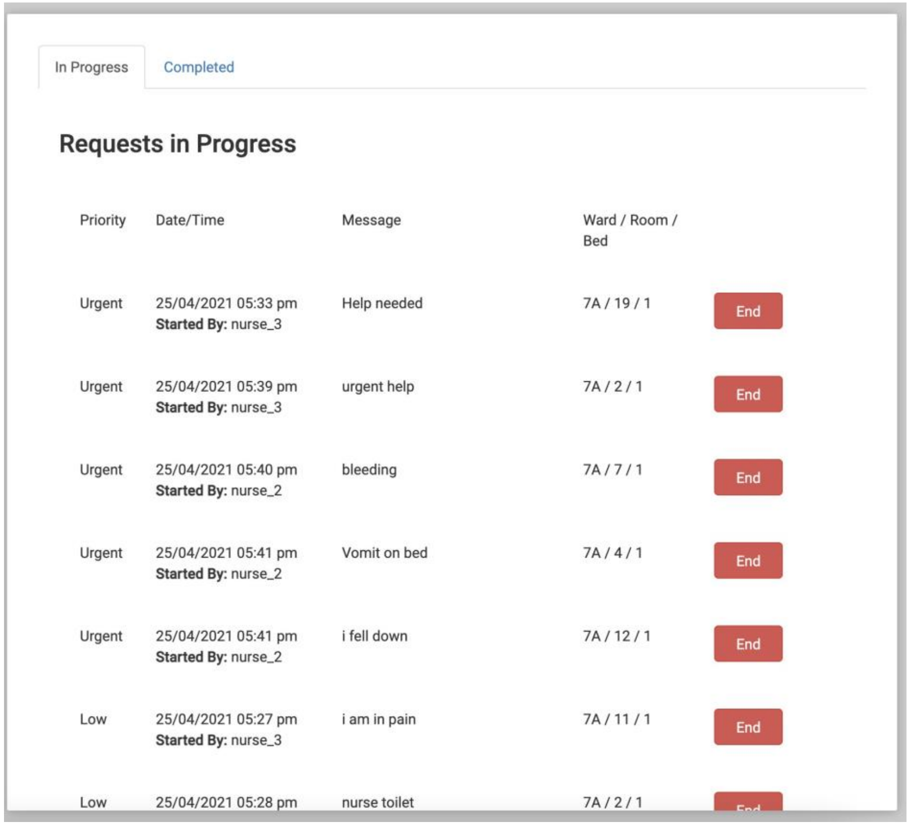


Figure 5: The Nurse ID input box that will appear once you start the task

The list of new requests is sent from the 'Query API' and is refreshed every 15 seconds. To select the requests, the nurse will simply click on the 'Start' button and input in his/her ID to attend to the request.



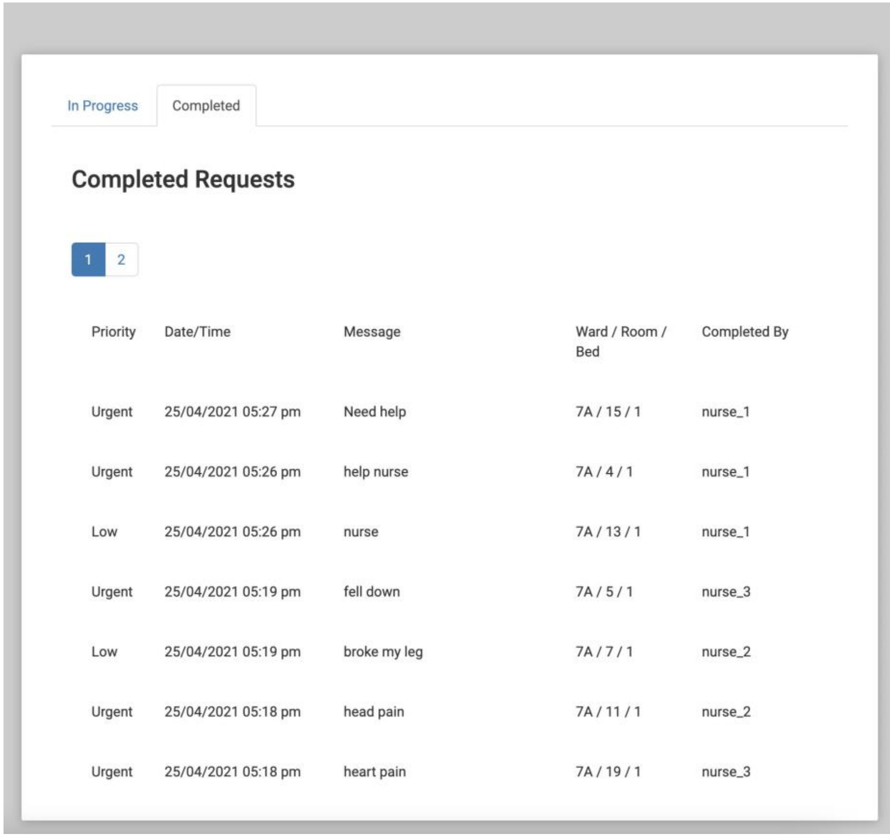
In Progress | Completed

Requests in Progress

Priority	Date/Time	Message	Ward / Room / Bed	
Urgent	25/04/2021 05:33 pm **Started By:** nurse_3	Help needed	7A / 19 / 1	End
Urgent	25/04/2021 05:39 pm **Started By:** nurse_3	urgent help	7A / 2 / 1	End
Urgent	25/04/2021 05:40 pm **Started By:** nurse_2	bleeding	7A / 7 / 1	End
Urgent	25/04/2021 05:41 pm **Started By:** nurse_2	Vomit on bed	7A / 4 / 1	End
Urgent	25/04/2021 05:41 pm **Started By:** nurse_2	i fell down	7A / 12 / 1	End
Low	25/04/2021 05:27 pm **Started By:** nurse_3	i am in pain	7A / 11 / 1	End
Low	25/04/2021 05:28 pm	nurse toilet	7A / 2 / 1	End

Figure 6: The ‘Requests in Progress’ table

Once the task is started, the request will be moved to the ‘Requests in Progress’ table, which is refreshed every 60 seconds. Once the request has been completed, the nurse will come back to the Nurse Station Panel and click on the ‘End’ button and input in his/her ID to end the request.



Priority	Date/Time	Message	Ward / Room / Bed	Completed By
Urgent	25/04/2021 05:27 pm	Need help	7A / 15 / 1	nurse_1
Urgent	25/04/2021 05:26 pm	help nurse	7A / 4 / 1	nurse_1
Low	25/04/2021 05:26 pm	nurse	7A / 13 / 1	nurse_1
Urgent	25/04/2021 05:19 pm	fell down	7A / 5 / 1	nurse_3
Low	25/04/2021 05:19 pm	broke my leg	7A / 7 / 1	nurse_2
Urgent	25/04/2021 05:18 pm	head pain	7A / 11 / 1	nurse_2
Urgent	25/04/2021 05:18 pm	heart pain	7A / 19 / 1	nurse_3

Figure 7: The ‘Completed Requests’ tab

There is also a ‘Completed Requests’ tab so that the nurse can view the list of completed requests in the last 24 hours.

All the data will be stored in the centralized database for data analysis for future refinement of the system.

5.1.3 Nurse Activity Manager (Admin Dashboard)

The Admin Dashboard is available only for ‘Admin Users’. This page displays all the ‘Registered Nurses’ (shown in Figure 8) and ‘Admin Users’ (shown in Figure 9) for the hospital in two different tabs.

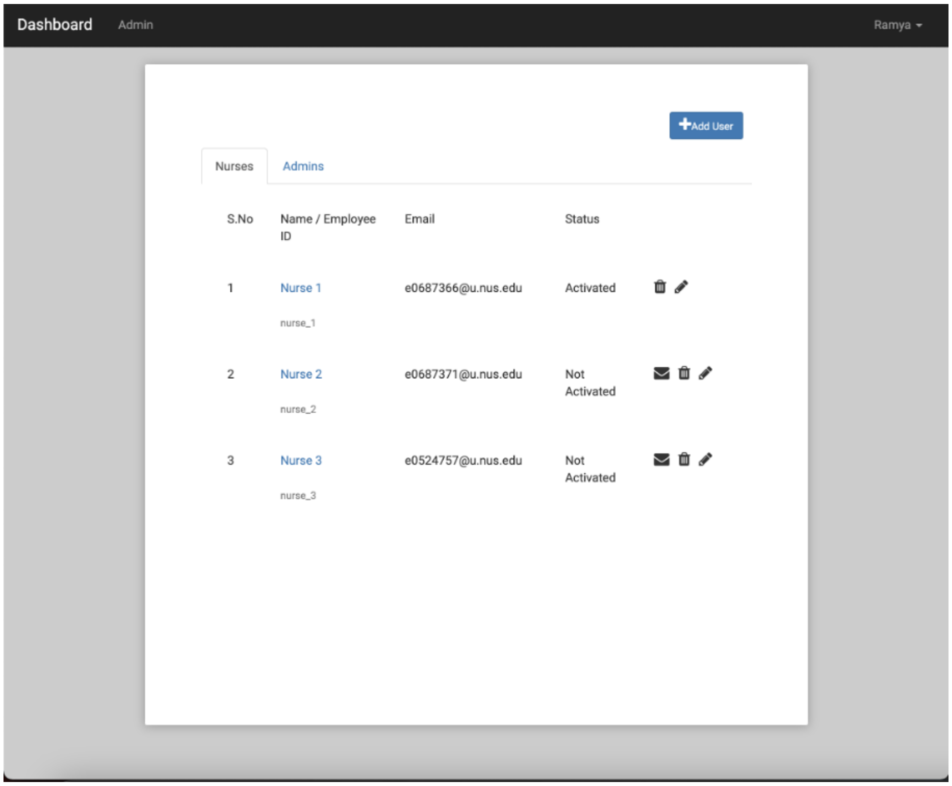


Figure 8: Screenshot showing the list of registered nurses.

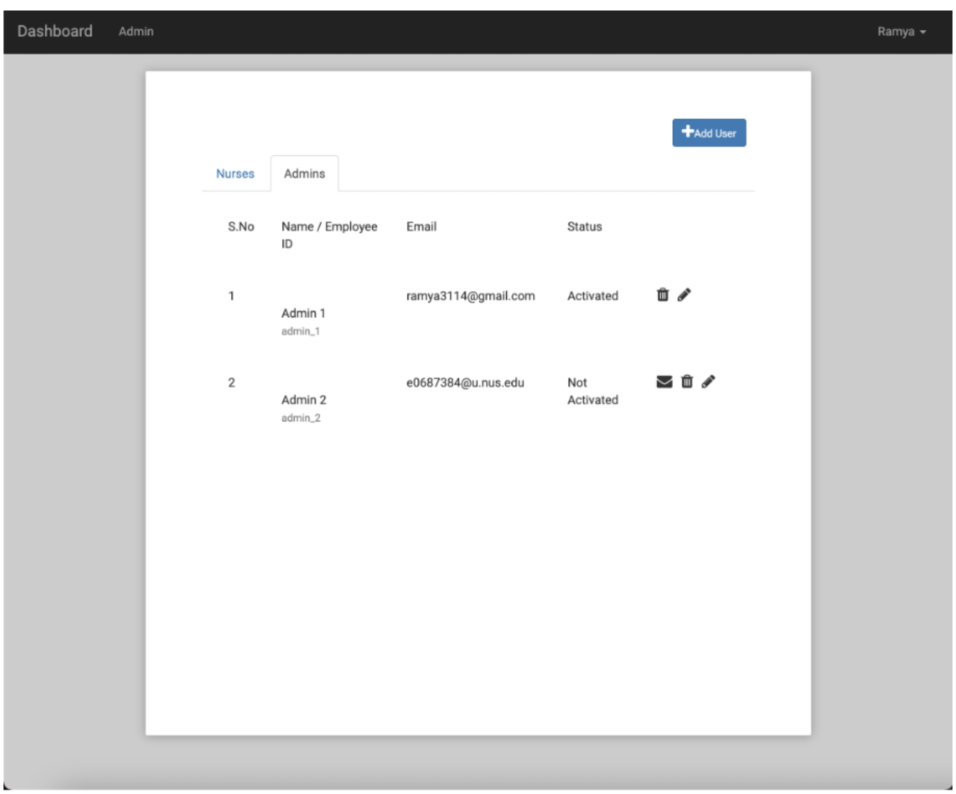


Figure 9: Screenshot showing all the admin users

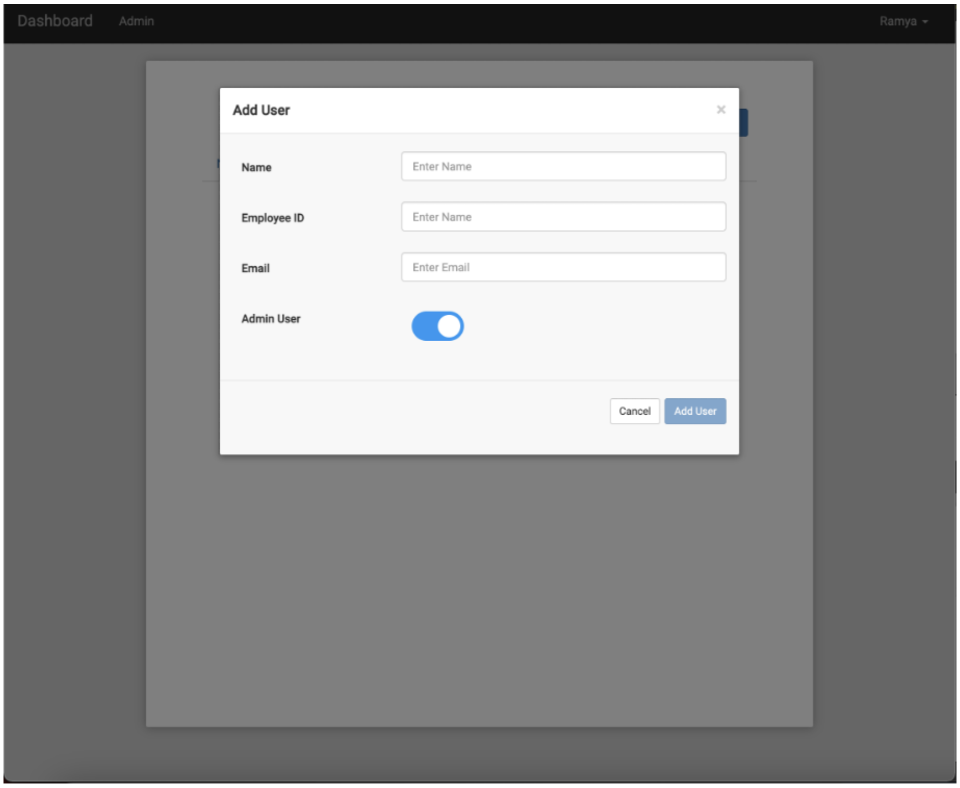


Figure 10: The ‘Add User’ function

The ‘Add User’ page allows the admin to add new users into the application using the ‘Add User’ button. When a new user is added, an email is sent to the given email id. The user must verify their account, to set a password and access the application by clicking on the link in the email, which expires within 24hrs. They must contact the admin to get a new verification email

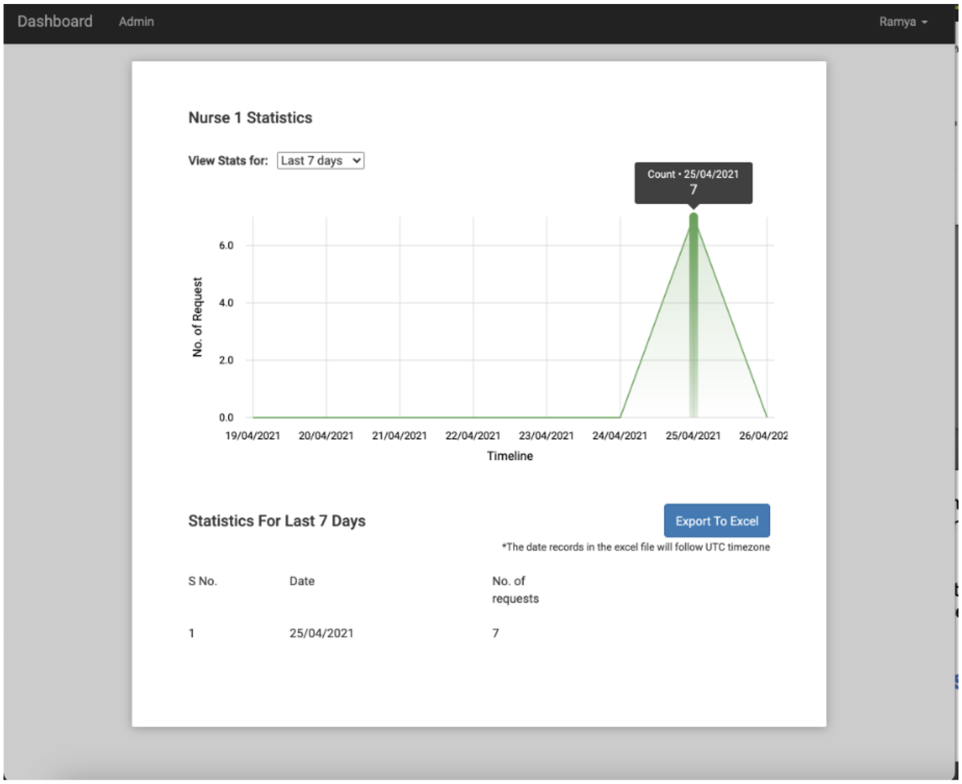


Figure 11: Screenshot showing the number of requests handled by the nurse over the past seven days.

Back In Figure 8, you can see that the name of each nurse is a hyperlink, which leads you to the statistics page as shown in figure 11. The statistics page displays the nurse's activities for the 'Day' in a graph and the table representation of the graph data.

The user has the option to get statistics for 'Past 7 days' or 'Past 30 days' as well. There is also an option to download the statistics as an Excel file.

5.2 Backend Processing

5.2.1 Classification of Request

Classification of patient requests is done through text classification. The process is shown in the diagram below.

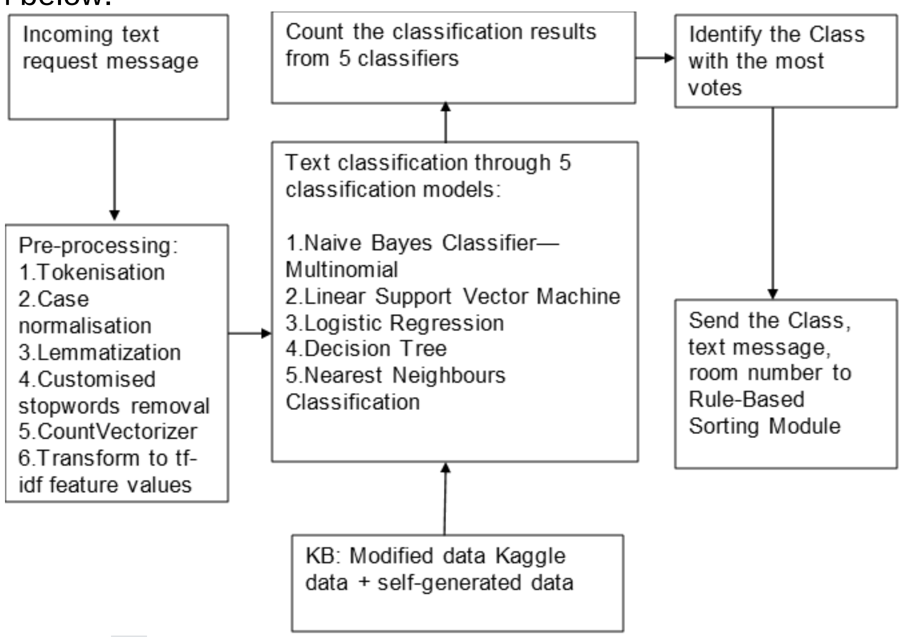


Figure 12:The Classification of the Request workflow

5.2.1.1 Machine Learning

A) Obtaining Training Data

One part of the training data is coming from Kaggle[1]--- Disease Symptom Prediction Data Set. The rating system in the training data is mainly used to define the seriousness of symptoms and therefore modification is needed to suit our system classification requirement. The rating is modified into 3 classes and punctuations are removed.

As this data set does not include non-medical requests which are commonly seen in patient's request. Some data is self-generated which covers the common non-medical request and other requests which is not covered in the Kaggle data set.

The data are stored in excel file in the following format:

Symptoms	Class
I wetted my bed	2
I need to pee	1
I have heartache	3

- Class 1: low (non-urgent and less important request)
- Class 2: medium (non-urgent and more important request)
- Class 3: Urgent (urgent request which needs immediate attention)

This dataset acts as a knowledge base which will be used in training.

B) Text Pre-processing Before Training

The request received in the real situation is expected to be in text format as sent over by Mycroft.AI. Mycroft will be sending sentences without punctuation. The punctuations are removed in training data as well. Therefore, in the text pre-processing, there is no punctuation removal. The text-pre-processing process is shown in the flow chart below.

The texts are tokenized, changed to lowercase, lemmatized to basic forms. To increase the classification accuracy, the stop words list is customized to remove unnecessary and commonly repeated words in patients' requests such as want. Important urgency indicating words such as very is removed from the standard stop words list.

Before pre-processing	After pre-processing
I wetted my bed	Wet bed
I need to pee	Need pee
I have heartache	heartache

However, the pre-processing does not end at this stage. The classifiers cannot classify text directly. The data will have to be transformed to values before they can be classified properly.

The data is converted to a matrix of token counts by using CountVectorizer. The matrix will then be transformed into a normalized tf-idf feature values through td-idf transformer. Tf-idf means term-frequency times inverse document-frequency[2], it gives a good indication of weighting of the terms in a dataset and with terms that appears in many documents given less weight. The toolkits used here is scikit-learn library. The complete process of pre-processing is shown below:

C) Training of Data

The dataset will be sent to five different text classifiers:

1.Naive Bayes Classifier Multinomial	Implements the Naive Bayes algorithm for multinomially distributed data, works well on tf-idf vectors and requires a small amount of training data.
2.Linear Support Vector Machine	Linear model for classification, creates a line or a hyperplane which separates the data into classes. Implement using **SGDClassifier** (loss='hinge')
3.Logistic Regression	Uses the cross-entropy loss and class is set to multinomial.
4.Decision Tree	Create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features.
5.Nearest Neighbours Classification	Classification is computed from a simple majority vote of the nearest neighbors of each point. K=5 as dataset is small.

All the classifiers can be found in scikit-learn library.
A class will be generated from each classifier. The accuracy from the first four classifiers is around 80%. The Multinomial Naive Bayes Classifier, Linear Support Vector Machine, Logistic Regression and Decision Tree are supervised learning algorithms that are used for classification of classes.

The accuracy of the KNN classifier is around 60%. However, the KNN classifier is kept as it engages the data differently as it identifies the nearest neighbour during classification. Its accuracy could increase with increased dataset in the future versions. As the dataset is small, k is set to 5. Optimisation methods will be added in the future to optimise the KNN classifier.

To increase the accuracy of our system, we implemented a hybrid system. The result from each classifier is counted and the class with the highest votes will be sent to the next sorting module.

5.2.1.2 Performances of Different Classifiers

The performances of different Classifiers are shown in figures below.

```
accuracy 0.7899159663865546
              precision    recall  f1-score   support

           1       0.68      0.96      0.80        47
           2       0.93      0.71      0.81        35
           3       0.92      0.65      0.76        37

    accuracy                           0.79       119
   macro avg       0.84      0.77      0.79       119
weighted avg       0.83      0.79      0.79       119

[[45  0  2]
 [10 25  0]
 [11  2 24]]
Accuracy Score: 78.99159663865547 %
```

Figure 13:Multinomial NB classification model and its performance outcome

```
accuracy 0.8151260504201681
              precision    recall  f1-score   support

           1       0.75      0.91      0.83        47
           2       0.83      0.83      0.83        35
           3       0.93      0.68      0.78        37

    accuracy                           0.82       119
   macro avg       0.84      0.81      0.81       119
weighted avg       0.83      0.82      0.81       119
```

Figure 14:Linear Support Vector model and its performance outcome

```
accuracy 0.8319327731092437
              precision    recall  f1-score   support

           1       0.74      0.96      0.83        47
           2       0.94      0.83      0.88        35
           3       0.93      0.68      0.78        37

    accuracy                           0.83       119
   macro avg       0.87      0.82      0.83       119
weighted avg       0.85      0.83      0.83       119
```

Figure 15:Logistic Regression classification model and its performance outcome

accuracy 0.7983193277310925

	precision	recall	f1-score	support
1	0.72	1.00	0.84	47
2	0.87	0.77	0.82	35
3	0.91	0.57	0.70	37
accuracy			0.80	119
macro avg	0.84	0.78	0.79	119
weighted avg	0.83	0.80	0.79	119

Figure 16:Decision Tree classification model and its performance outcome

accuracy 0.6386554621848739

	precision	recall	f1-score	support
1	0.64	0.79	0.70	47
2	0.55	0.69	0.61	35
3	0.88	0.41	0.56	37
accuracy			0.64	119
macro avg	0.69	0.63	0.62	119
weighted avg	0.69	0.64	0.63	119

Figure 17:KNN Classifier model and its performance outcome

5.2.1.3 Machine Reasoning

After the dataset is trained, new data generated from Mycroft.AI will be sent through the trained model and the classification class will sent to rule-based sorting module. The process flow is shown below:

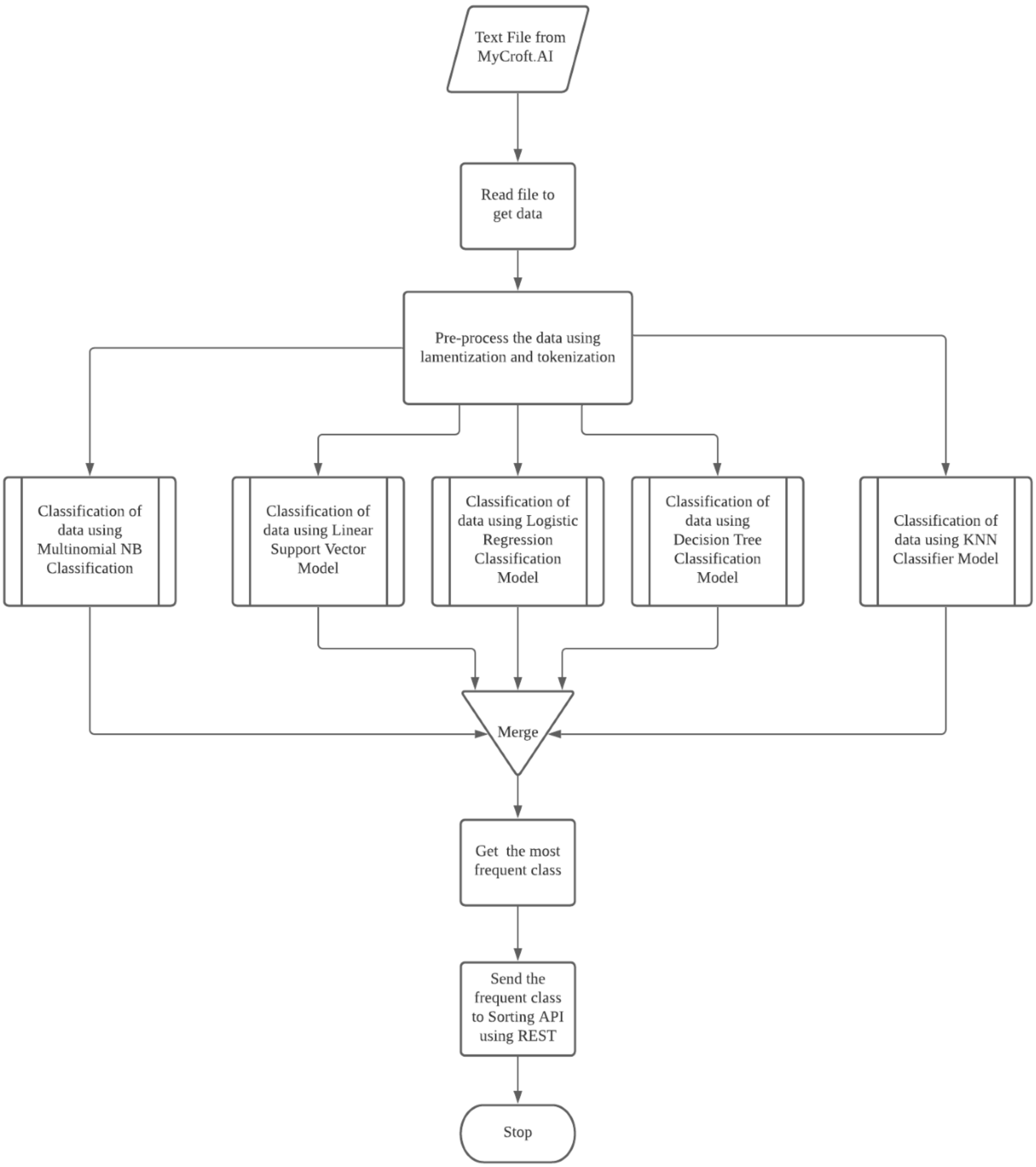


Figure 18: Workflow of Classification Process

5.2.1.4 Toolkits Used

Pre-processing	Natural Language Toolkit (NLTK)
	`from nltk.corpus import stopwords`
	`from nltk.corpus import wordnet`
	`from nltk.stem import WordNetLemmatizer`
	`nltk.download('stopwords')`
	`nltk.download('punkt')`
	`nltk.download('averaged_perceptron_tagger')`
	`nltk.download('wordnet')`

Machine Learning	scikit-learn
	`from sklearn.metrics import accuracy_score, confusion_matrix, classification_report`
	`from sklearn.model_selection import train_test_split`
	`from sklearn.preprocessing import StandardScaler`
	`from sklearn.tree import DecisionTreeClassifier`
	`from sklearn.naive_bayes import MultinomialNB`
	`from sklearn.neighbors import KNeighborsClassifier`
	`from sklearn.feature_extraction.text import CountVectorizer`
	`from sklearn.feature_extraction.text import TfidfTransformer`
	`from sklearn.pipeline import Pipeline`
	`from sklearn.linear_model import SGDClassifier`
	`from sklearn.linear_model import LogisticRegression`

Others	NumPy, Pandas

5.2.1 Dynamic Rule-Based Sorting

The dynamic rule-based sorting module performs the following tasks:

a) Checks that requests that are still tagged with a ‘New’ status (i.e. not yet attended to by a nurse) do not exceed the waiting period of 20 minutes from the time the request was registered in the system. If a request has been unattended for more than 20 minutes, the highlight flag will be set to true. The highlight flag is generated post query and is not stored in the database.

b) Sorts the request record set against key considerations to ensure that the most urgent requests are shown at the top of the request queue. The records are sorted in terms of:

1) Highlight flag whereby those that are highlighted are sorted to the top since they need attention as soon as possible.
2) Priority level of the request with the most urgent requests (i.e. those with priority value of High are moved to the top of the queue).
3) The date and time the request was made and records with earlier date and time will be sorted to the top.

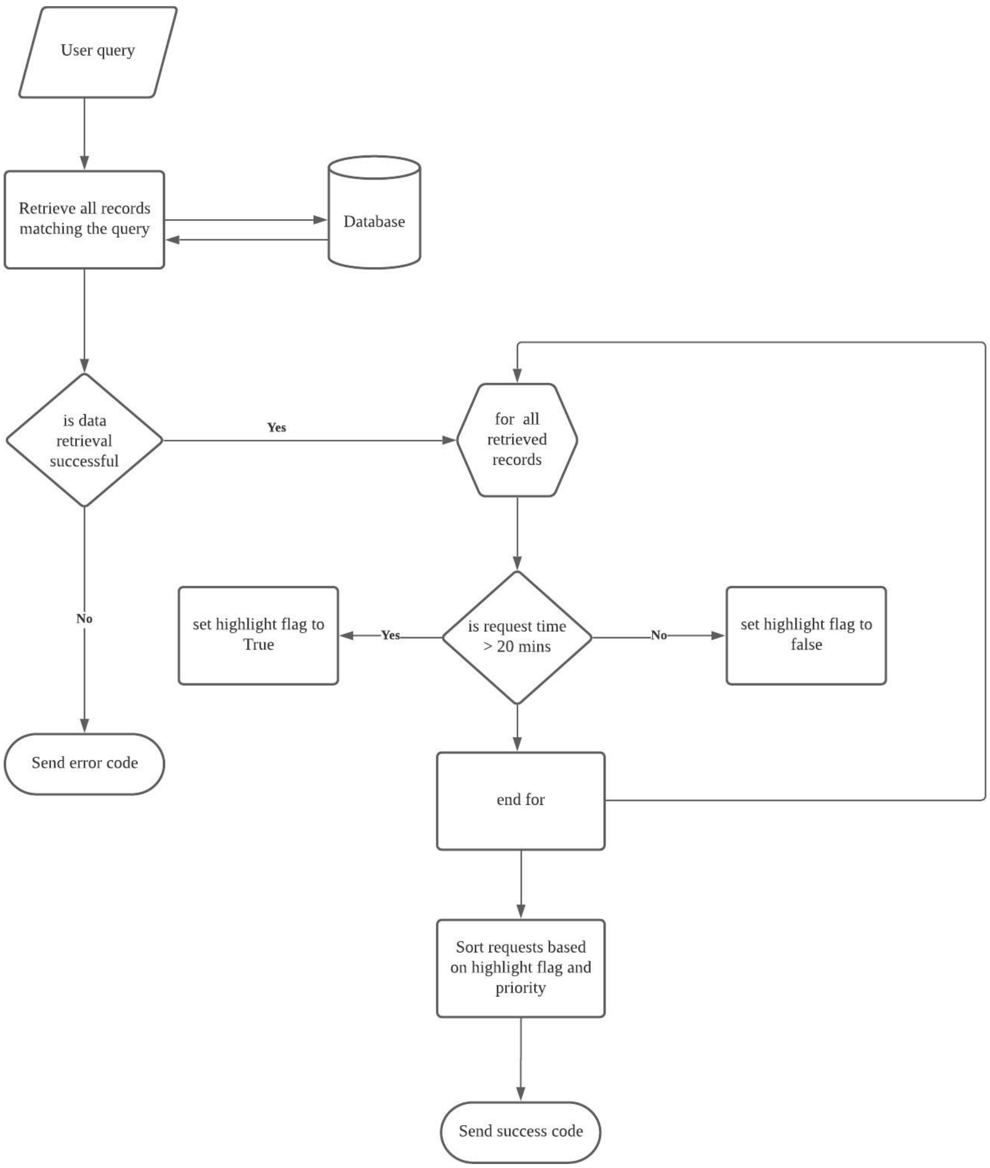


Figure 19: Workflow of the dynamic rule-based sorting module.

The dynamic rule-based sorting process ensures that the most urgent requests are always sorted to the top of the queue with those overdue requests (stuck for more than 20 minutes in the queue) moved to the head of the queue. This ensures that the nurses will have visibility to all requests and not leave any unattended.

5.2.2 User Creation and Authentication Processes

a) User Creation

A new user is created when an api call is made from frontend with user input. Once the api receives user input, it is then converted into a schema object using 'marshmallow' and where a default value is set for missing fields appropriately. A random verification code of 16 characters length is generated, for user email verification purpose, along with an expiration date for this code. This schema is then added into the 'Users' collection in mongoDB using pymongo.

After the user record is successfully inserted into the database, a success code is sent to frontend and simultaneously another process is invoked in the background to initiate sending an email to the user's email ID using SendGrid email template and APIs. This workflow is illustrated in Figure 20.

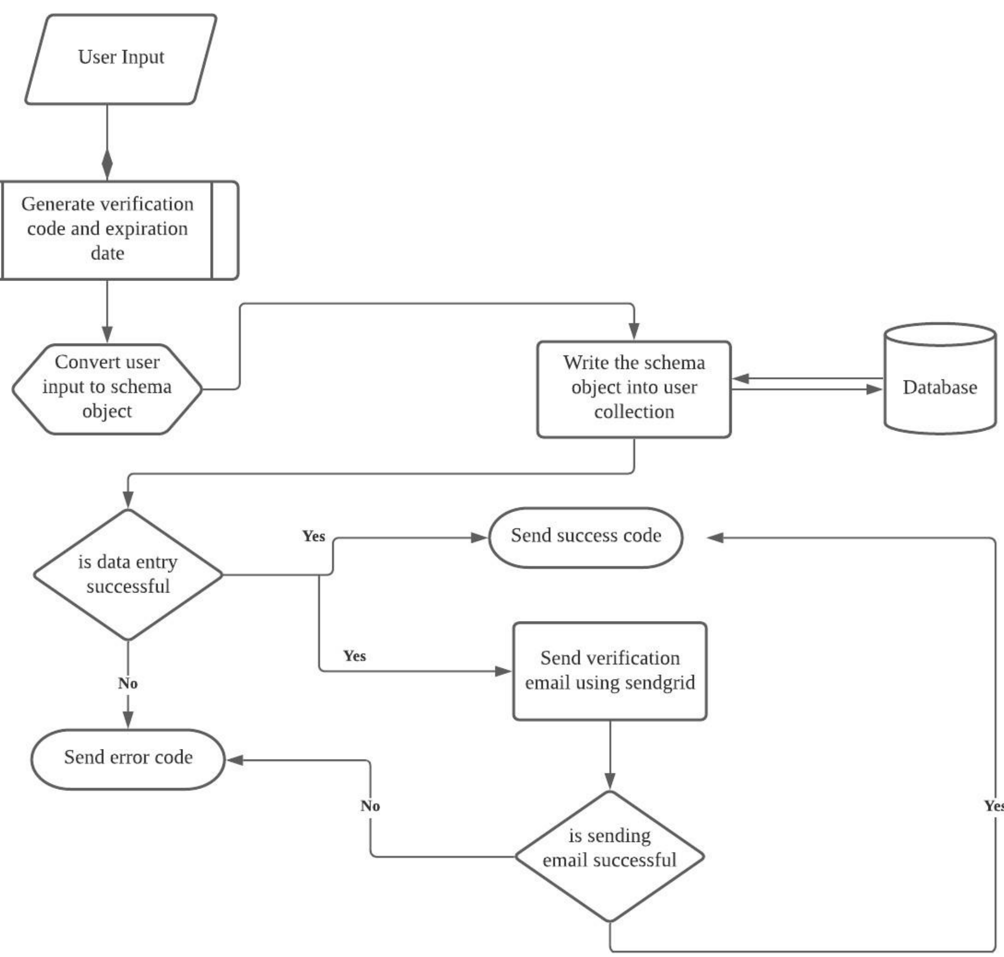


Figure 20: Workflow of the User Creation process

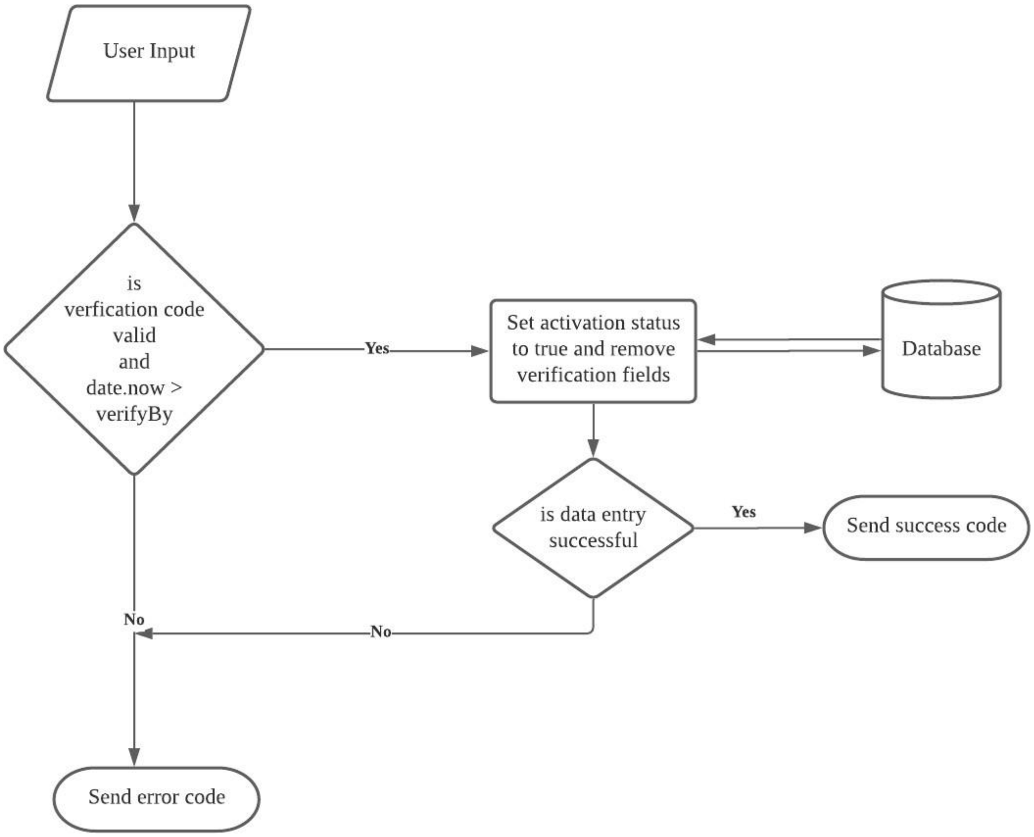


Figure 21: Workflow of the User Verification Process

When the user clicks on the verification email, the 'User Verification' API input is checked against the records in the collection and if the 'verifyBy' time is not greater than current time, to retrieve the record the matching record. If a matching record exists, then the activation status is set to true, and the user is redirected to the 'set password' page.

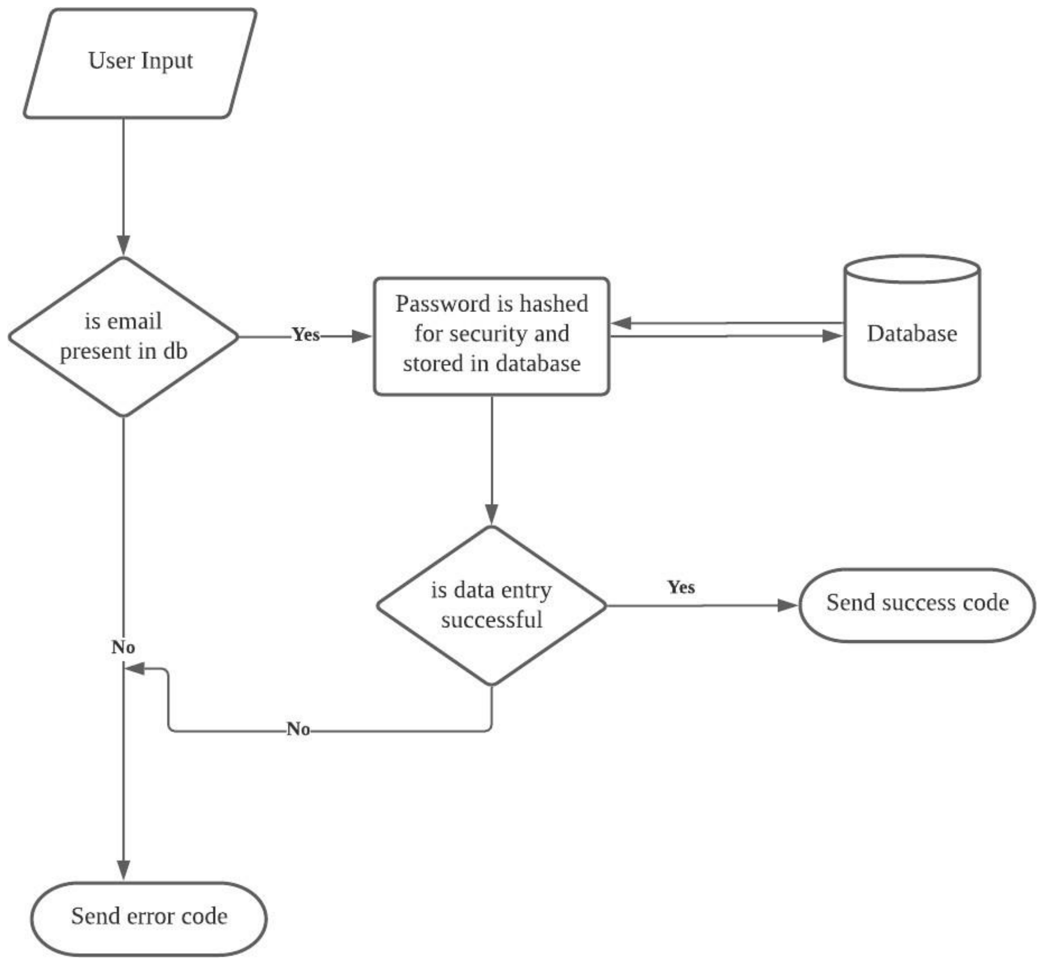


Figure 22: Workflow of the Password Creation Process

When a user enters his/her password into the form and clicks on the ‘Submit’ button. The create password API will be called, the email from user input is checked against the database to see if the user exists and the matched record is retrieved. If a matched record is retrieved, the input password is encrypted by a hashing algorithm to enhance security and then stored into the database.

b) Authentication Process:

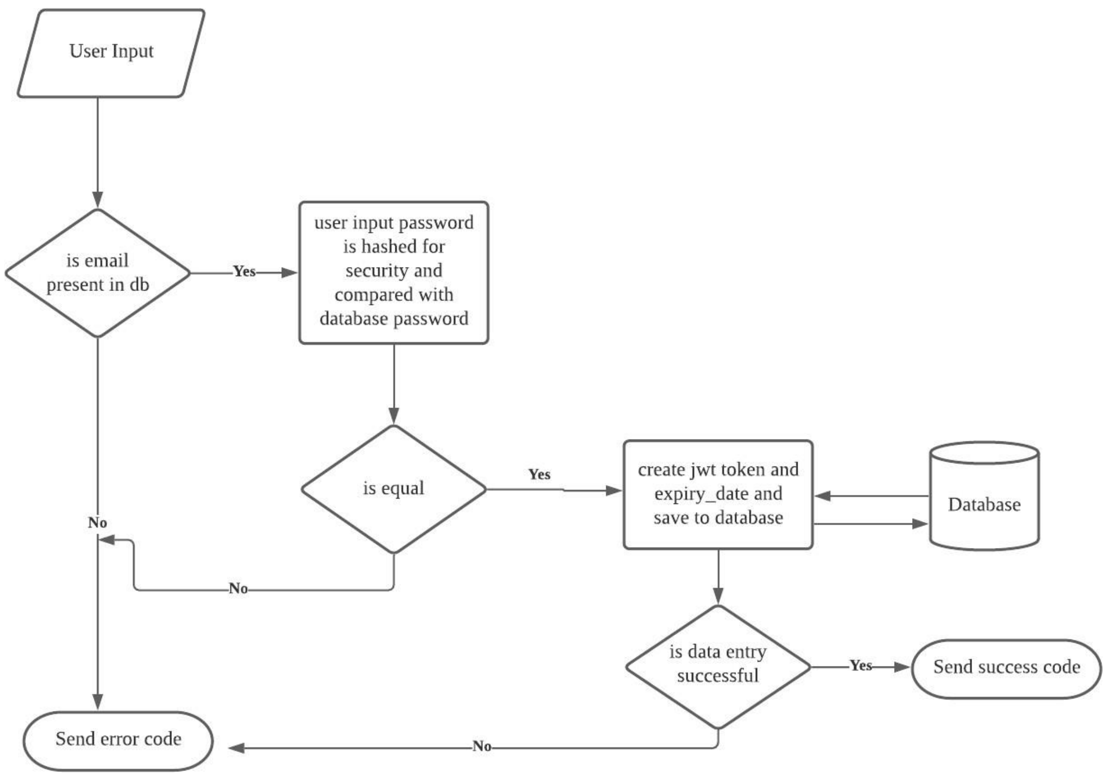


Figure 23: Workflow of the Login Process

When the login api is called with email and password input from the frontend, the email is checked against the database to see if the user exists. If the user exists, then the hashed password from the database is compared against the user input password using bcrypt function, to see if both matches. If both passwords match, then a jwt token is created with 8 hours validity and is saved into the 'Sessions' collection and is also returned to the frontend.

5.3 Data Management

5.3.1 Storage of data

The data in our system is stored in the MongoDB database system. The reasons for choosing MongoDB are as follows:

- Both cloud and local installations are available.
- It allows the application to be built more quickly.

- Very good in handling highly diverse data types compared to more traditional relational databases.
- The array-type structure of the document allows one to map the data naturally to modern day object-oriented programming languages.
- Highly scalable to meet the needs of the application.

The Hey Nurse! database is named ‘PatientRequests’ and the data is stored as JSON-formatted documents in the collections within the database. A collection is similar in concept to a table in relational database. Figure 24 illustrates the collections within the database and the name of the labels or keys for the value stored.

The ‘Requests’ collection stores the requests from the patients and as well as the location where the request was made (i.e., the ward, room and bed number). The request priority (or classification) generated by the NLP classification module is also captured together with the date/time the request was sent from the Mycroft API to the NLP classification module. Finally, the nurse ID, the start and end date/time of a nurse attending to the request together with the various states (a.k.a. status) of the request are also stored in the same ‘Requests’ collection.

The ‘Users’ collection stores the information associated with each user that is registered or created in the system. A user can have either one of the two roles, an Administrator, or a regular user. Currently, the system assigns nurse manager and system administrator to the Administrator role while all other nurses are assigned the regular user role. This is achieved by setting the ‘isAdmin’ flag accordingly. The other information in this table pertains to the login credential of the user and whether the credential has been verified.

The ‘Sessions’ collection is used by the system to manage the session for each user. Since this is a web-based application which is stateless, the session information related to the user is important. They provide the application the ability to navigate between pages and, at the same time, retain knowledge of actions taken by that user in the preceding page(s). More collections can be added to the database depending on the user requirements.

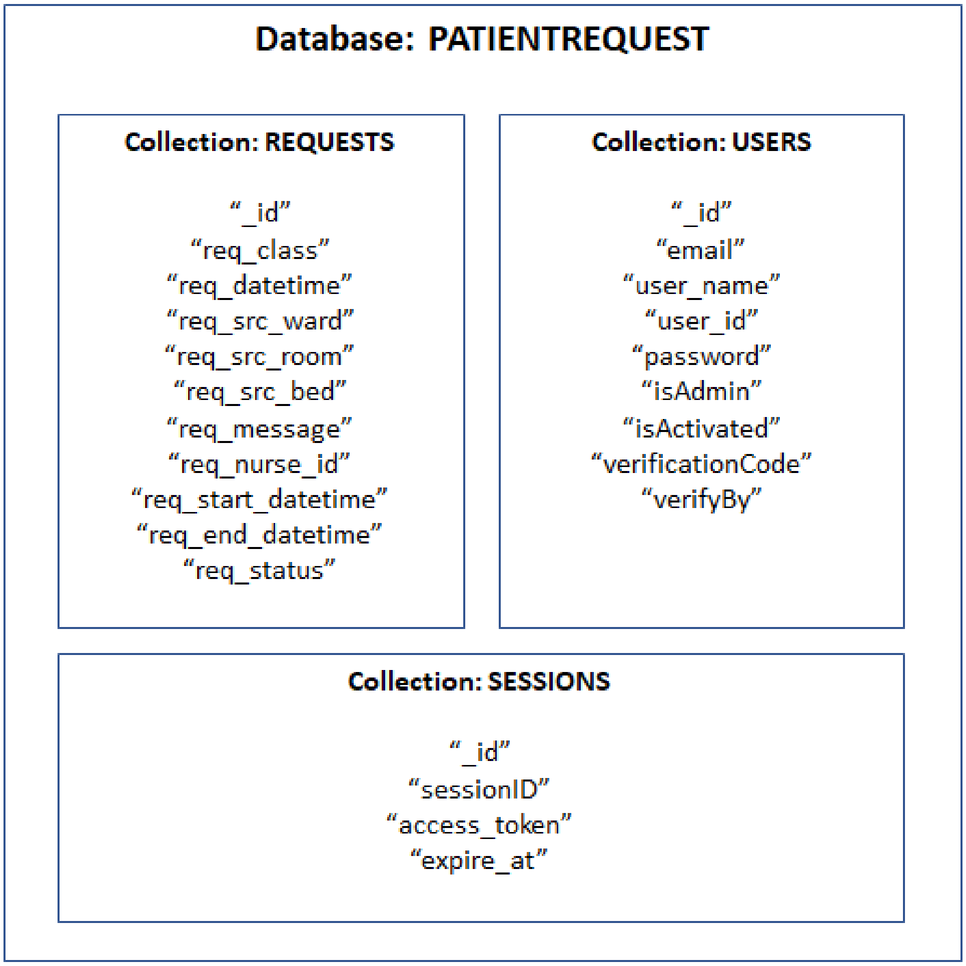


Figure 24: Schema Representation of the PATIENTREQUEST database

5.3.2 Data Relationship

As described in 5.3.1, the data from the system is stored in the 'Requests', 'Users' and 'Sessions' collections. The relationships between the collections are shown in the diagram below.

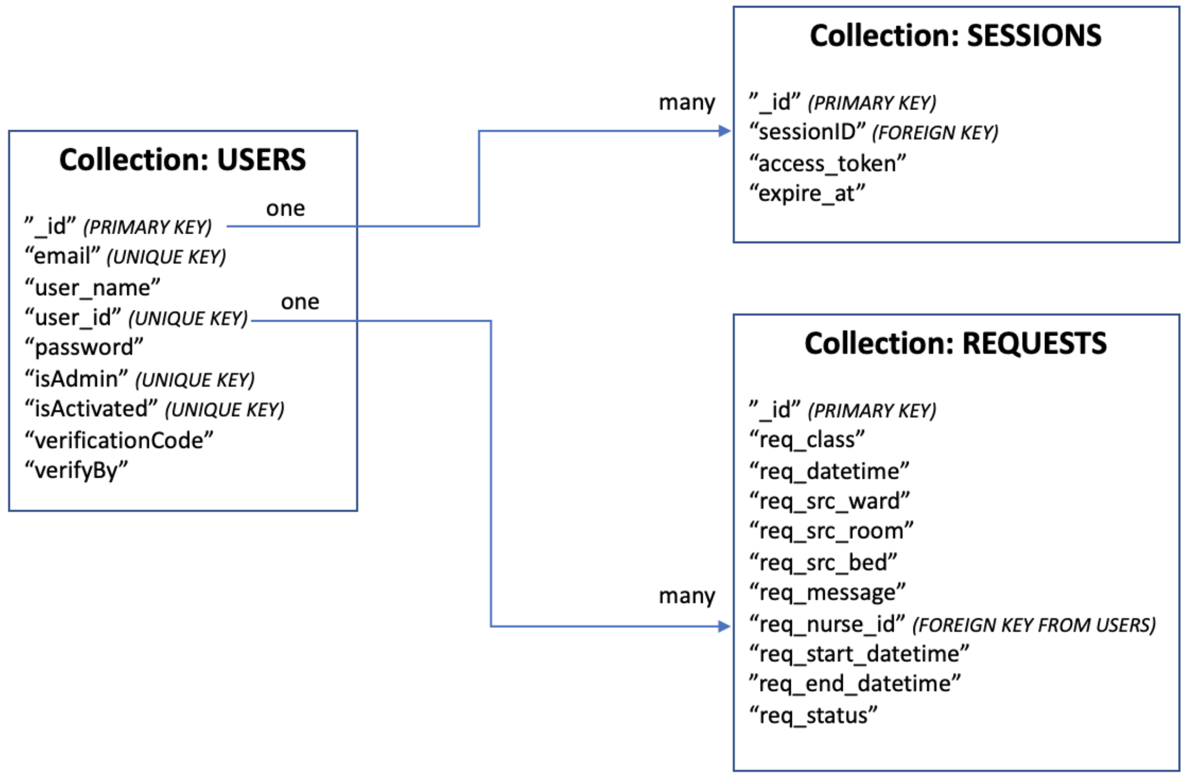


Figure 25: Relationship diagram of the three Collections.

The system is user centric in that all key information is related to a specific user. This can be explained by the examples below.

1. The nurse is a user by definition and all requests that are attended to are associated with a particular nurse via the 'Users.user_id' and 'Requests.req_nurse_id' relationship. Each nurse can have more than one requests that he/she is associated with.

2. In a separate example, a nurse manager is also a user and when he/she queries the system for analytic insight into the nurses' activities, it is also via the same 'Users.user_id' and 'Requests.req_nurse_id'.

3. The 'Users._id' to 'Sessions.sessionID' relationship defines that a user can have many sessions since he/she will log on to the system at different times and each login will spawn a new session. The session data is used to manage user interaction with the system and to preserve the state of connection between the web browser and the server.

6 Conclusion

Our team was successful in deploying the Hey Nurse!. Of course, this comes with its own set of challenges and there are also many ways that we can improve the implementation of the system with the luxury of more time and resources.
Below describe the challenges we faced, how we tackled them and lastly suggested improvements for future upgrades.

6.1 Assumptions

- The deployment will be a one-to-one Mycroft device besides each patient bed. There will be a Centralized Nurse Station on that floor to a population of 20 patients.
- Patients who cannot speak will not be able to use the system. They must use the traditional call bell system.
- English language to be the only medium as of now. The wake word is ‘Christopher’

6.2 Challenges & Future Developments

- **Environmental Noise in the Ward.** The environmental noise may affect the device’s ability to pick up the patient’s voice correctly and will affect the voice-to-text accuracy. We will need more data to fine-tune the system and may look into other better Voice-to-Text software’s such as Nvidia Jarvis to tackle the issue.
- **Applications in other healthcare sectors.** Although this is currently a hospital environment deployment, our system can be easily deployed in other areas such as Senior Care Centers, Psychiatrist facilities etc.
- **Manual keyboard input for the Nurse ID.** In the future, we will have a Card scanner to capture nurse ID from the employee card.
- **Nurse activity record is limited**. We will expand the database to include participation of multiple nurses for completing a request.
- **Emotion sensing is not active.** In the future, we could use Knowledge Graph or Neural Network to improve on emotion sensing.
- **Conversational limitations on Mycroft.** Currently there is no real time feedback to the patients’ requests. In future, we hope to improve Mycroft to provide a conversation based on the patient's request.

- **Reaching out to the Public.** At this current stage in time, our team does not have the time or resources to market or promote our system in order for it to gain traction among a wider audience. However, we believe that by word of mouth and the prevalence of social media would help us to boost the awareness of our product.

APPENDIX A - Project Proposal

GRADUATE CERTIFICATE: Intelligent Reasoning Systems (IRS)
PRACTICE MODULE: Project Proposal

Date of proposal: 2 May 2021
Project Title: HEY NURSE! an Intelligent Nurse Call System
Sponsor/Client: *(Name, Address, Telephone No. and Contact Name)* Institute of Systems Science (ISS) at 25 Heng Mui Keng Terrace, Singapore NATIONAL UNIVERSITY OF SINGAPORE (NUS) Contact: Mr. GU ZHAN / Lecturer & Consultant Telephone No.: 65-6516 8021 Email: zhan.gu@nus.edu.sg
Background/Aims/Objectives: The objective of our project is to design and integrate an intelligent system, which will help nurses to respond to patients in their ward more efficiently by providing recommendations prior to nurses picking up the requests from their workstation.
Requirements Overview: 1. Knowledge extraction from dataset 2. Programming ability – Python, Angular 3. Angular framework understanding – components, services, guards 4. System integration ability between Angular, MongoDB and Python
Resource Requirements (please list Hardware, Software and any other resources) Hardware proposed for consideration: - CPU Software proposed for consideration: - Mycroft.AI - Linux OS (Ubuntu/Debian)

- VirtualBox (to install Ubuntu, if other OS environments are used.) with at least 7GB memory assigned

Number of Learner Interns required: (Please specify their tasks if possible)

A team of four project members

Methods and Standards:

Procedures	Objective	Key Activities
Requirement Gathering and Analysis	The team should meet with ISS to scope the details of the project and ensure the achievement of business objectives.	1. Gather & Analyse Requirements 2. Define internal and External Design 3. Prioritize & Consolidate Requirements 4. Establish Functional Baseline
Technical Construction	To develop the source code in accordance with the design. To perform unit testing to ensure the quality before the components are integrated as a whole project.	1. Setup Development Environment. 2. Understand the System Context, Design. 3. Perform Coding. 4. Conduct Unit Testing.
Integration Testing	To ensure interface compatibility and confirm that the integrated system hardware and system software meets the requirements and is ready for acceptance testing.	1. Prepare System Test Specifications. 2. Prepare for Test Execution. 3. Conduct System Integration Testing. 4. Evaluate Testing. 5. Establish Product Baseline.
Acceptance Testing	To obtain ISS user acceptance that the system meets the requirements.	- Plan for Acceptance Testing. - Conduct Training for Acceptance Testing. - Prepare for Acceptance Test Execution. - ISS Evaluate Testing.
Delivery	To deploy the system into a production (ISS standalone server) environment.	● Software must be packed in accordance with ISS standards. ● Deployment guidelines must be provided in ISS production (ISS standalone server) format. ● Production (ISS standalone server) support and troubleshooting process must be defined.

Team Formation & Registration

Team Name: FARY
Project Title (repeated): HEY NURSE! an Intelligent Nurse Call System
System Name (if decided): HEY NURSE

Team Member 1 Name: Yap Jing Yang
Team Member 1 Matriculation Number: A0229976H
Team Member 1 Contact (Mobile/Email): +65-98471205 / e0687384@u.nus.edu

Team Member 2 Name: Francis Tan Wee Ming
Team Member 2 Matriculation Number: A0229963N
Team Member 2 Contact (Mobile/Email): +65-81136380/ e0687371@u.nus.edu

Team Member 3 Name: Ramya Shanmugam
Team Member 3 Matriculation Number: A0229958H
Team Member 3 Contact (Mobile/Email): +65-93366404 / e0687366@u.nus.edu

Team Member 4 Name: Wang Dongchen
Team Member 4 Matriculation Number: A0083129E
Team Member 4 Contact (Mobile/Email): +65-91218158 / e0524757@u.nus.edu

For ISS Use Only		
Programme Name:	**Project No:**	**Learner Batch:**
Accepted/Rejected/KIV:		
Learners Assigned:		
Advisor Assigned: Contact: Mr. GU ZHAN / Lecturer & Consultant Telephone No.: 65-6516 8021 Email: zhan.gu@nus.edu.sg		

APPENDIX B

Mapped System Functionalities against knowledge, techniques and skills of modular courses

Modular Courses	System Functionalities / Techniques Applied
Machine Reasoning (MR)	**Knowledge Representation** **Rule Based System** **Supervised Learning Algorithms**
Reasoning System (RS)	**Different classification models to recommend the best priority level for the request.**
Cognitive System (CGS)	**Cognitive Systems**: NLP & Mycroft API **Intent Training & Entity Definition:** Training phrases for all intents.

APPENDIX C

Installation and User Guide

Refer to the following GITHUB link: https://github.com/ramya0311/IRS-PM-2021-01-16PT-GRP-FARY-HeyNurse/tree/master/ProjectReport for installation and user guide.

APPENDIX D - Individual Reports

Refer to the following GITHUB link: https://github.com/ramya0311/IRS-PM-2021-01-16PT-GRP-FARY-HeyNurse/tree/master/ProjectReport for the individual team member report

APPENDIX E – References

References

1. Gek Phin Chua, Asia Pac J Oncol Nurs. 2020 Jul-Sep; 7(3): 259–265.

2. Mycroft.AI Installation and Skills development: https://mycroft-ai.gitbook.io/docs/

3. MongoDB using python:
   - Pymongo: https://pymongo.readthedocs.io/en/stable/tutorial.html
   - Marshmallow: https://marshmallow.readthedocs.io/en/stable/
   - MongoDB installation: https://docs.mongodb.com/manual/

4. Basic project setup reference: https://developer.okta.com/blog/2019/03/25/build-crud-app-with-python-flask-angular

5. Angular development: https://angular.io/docs

6. Development issues and solutions: https://stackoverflow.com/