NANYANG TECHNOLOGICAL UNIVERSITY

NANYANG TECHNOLOGICAL UNIVERSITY

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

CZ2006 BCS2 SOFTWARE ENGINEERING

Project Group: ALPHALAB

HealthLink

Team Members:

Sai Shein Htet (U2021780B)

Yap Yu Xiang, Shawn (U2021103J)

Sng Yi Xuan (U2021009B)

Ivan Lua Yangzhi (U2022506F)

Venkataraman Sidhaarth (U2021808J)

Zhou Qiren (U2021125J)

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
Shawn, Qiren, Shein Htet, Sidhaarth, Ivan, Yi Xuang	11/04/22	Checking of report	V1

1. Introduction

1.1 Purpose

The purpose of the software requirement specification is to provide supportive information on HealthLink. The document will also explain the purpose, features, and interface of the application. Furthermore, the document will also present a detailed description about the operating environment, design and implementation constraints of HealthLink.

1.2 Document Conventions

Conventions	Font Size	Font Type	Font Style
Headings	18	Times	Bold
Sub Headings	14	Times	Bold
Sub-sub Headings	11	Times	Italic
Normal Text	11	Arial	None
Important Names	11	Times	Bold

1.3 Intended Audience and Reading Suggestions

This software requirement document is intended for developers, testers, clients and users.

Developers who can review the application capabilities and easily understand which section of the application should be targeted in order to add new features or improve the capabilities of the application for future enhancement. Developers can view overall description, system features and other nonfunctional requirements for more understanding of the application.

Testers who want to use this document as a guideline for testing strategy as bugs are easier to discover by using a requirement document. This way of testing becomes more structured and organized. Testers can view product perspective, design and implementation constraints, system features, safety requirements and security requirements for base gathering of testing.

Clients who want to check if the desired outcomes have been achieved can refer to this document for verification. Clients can view product functions, user classes and characteristics, external interface requirements, system features and other non-functional requirements to verify if their requirements have been met.

Users who are interested in reading the software requirement document to understand what is the main functionality of the application. Users can view product functions and user interfaces for the features of the application. User can also refer to the user manual for a step-by-step process of how to use the application

1.4 Product Scope

HealthLink aims to provide an integrated solution to improve the efficiency of Singapore's healthcare system. We aim to transform the user journey of obtaining healthcare in Singapore by creating an integrated solution that provides users a seamless user experience from the start of their user journey, where one gets a diagnosis, till the end where the user joins the queue of a selected healthcare facility. The process will be captured in one smooth connected flow, establishing a new, smart and innovate way of obtaining healthcare in the future

Efficient allocation of medical resources
Our web app aims to allow users to obtain a preliminary diagnosis based on suspected symptoms. By relying on the preliminary diagnosis, the users then can make a more informed decision on how they should tap on public healthcare resources, thus reducing the burden on the healthcare facilities, enhancing the quality of healthcare services.

Integration of information for better management
Our web app integrates the currently existing dispersed information of healthcare facilities in Singapore into one integrated and collective application that allows users to easily view, obtain and decide for themselves the best option to take.

By including information of healthcare providers such as queue length, the app aims to enhance the quality of central management of the healthcare system. Through increasing the transparency of the users' journey, more can be done to improve on their experience. For example, by monitoring the queue length of the healthcare providers, direct patients to less crowded locations.

Improvement of the quality of information
By providing an authoritative source of medical information and suggestion, the app aims to reduce the reliance on unverified information such as internet prescription and anecdotal medical suggestions. It also prompts the reluctant patients to take timely action against their worsening health conditions, thus improving the health of the population.

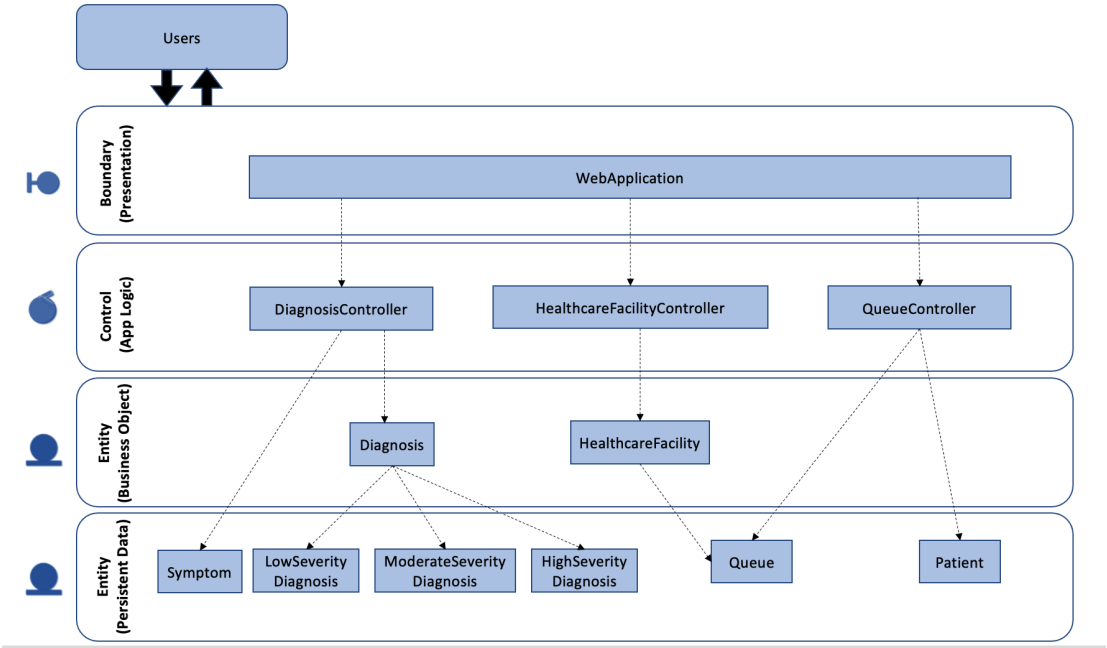
Facilitating joining queue from the user's perspective
Our web app aims to allow users to easily join the queue of any public hospital in Singapore.

After one has joined the queue, they would also be able to track the status of the queue directly from the application. Useful information such as the hospital name, queue number and number of people ahead of the queue will be provided.

2. Overall Description

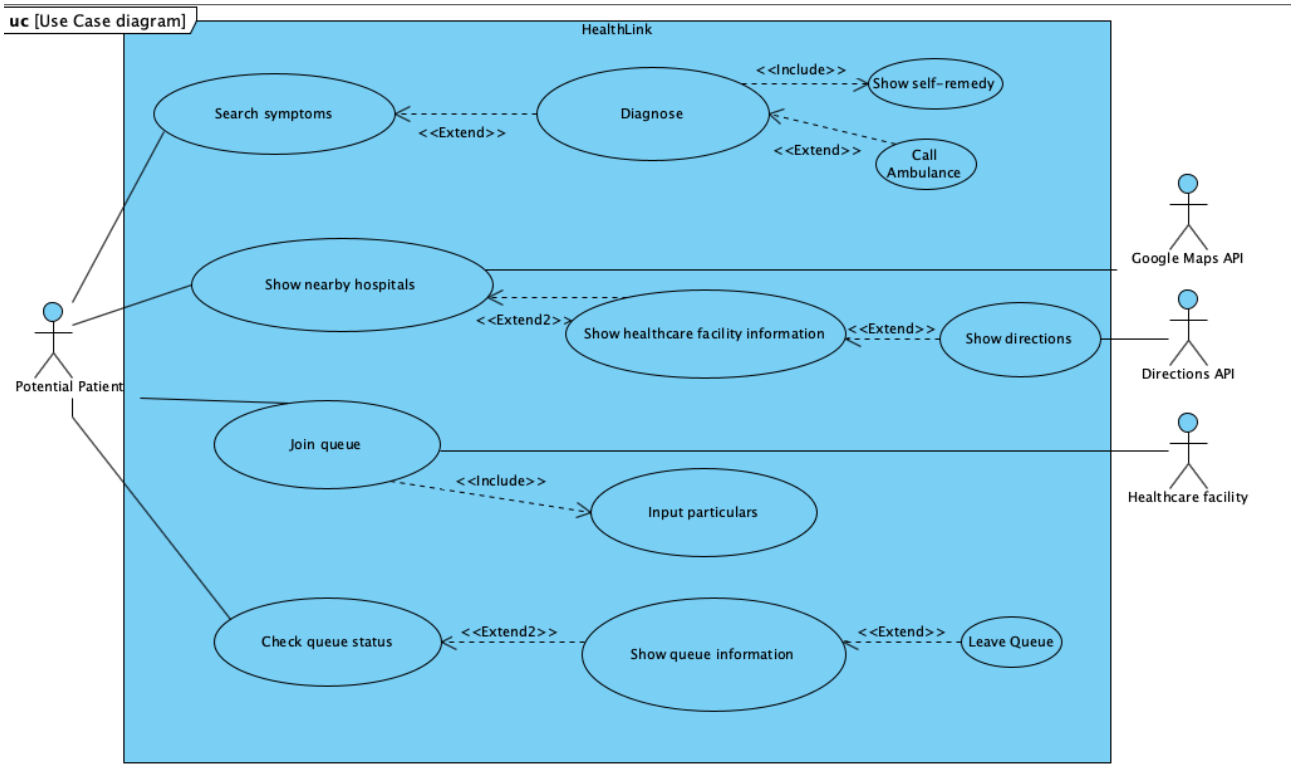
2.1 Product Perspective

HealthLink is a one stop real time diagnosis, nearest healthcare facility finder and navigator, and queue management system all rolled into one, a utility tool created for healthcare facilities and potential patients. HealthLink houses access to a centralised database of queues of different hospitals, and a database of symptoms and the possible diagnoses. The app grants real time updates to users on the current situation of the queue at a particular hospital's queue that they have joined, increasing time efficiency and reducing crowding at the healthcare facility. The built-in diagnosis tool gives the user a better perspective into the type and level of care needed. Ultimately, HealthLink empowers the users and medical professionals to make better and more effective and informed healthcare decisions.



This diagram shows a 4-layered system architecture. The top layer represents the presentation, which is a user interface that the user interacts with. The second layer is representative of the app logic which are basically the function calls. The third layer represents the entities that are business objects, such as diagnosis and healthcare facilities. The final and fourth layer is the persistent data which is where our app collects and stores the information which can be accessed and updated with ease.

2.2 Product Functions



The system involves 3 main modules, diagnosis, nearby healthcare facilities, and check queue status.

The first module diagnosis allows the users to input and search their symptoms. After that, based on the combination of symptoms they have, the module leverages on a machine learning model to provide a preliminary diagnosis, and further courses of action. The possible preliminary diagnoses are further split into 3 categories, low, medium and high severity to signal to users the level of healthcare support they should get.

The second module, has a location and navigation focussed basis, and shows the nearest healthcare facilities that users can go to, either based on a provided postal code or the user's current location. The user can access details about each healthcare facility, as well as the shortest path there. This links to the third module, which is the queue management system. The user can select one of the healthcare facilities and choose to join that facility's queue remotely.
The third module is comprised of the queue management system. The user enters their personal particulars, and the system is able to show their current place in the queue and the number of people in front of them. There is a check queue function as well which allows the user to check their status in real time. In summary, the system aims to walk users through the entire process of healthcare, from preliminary diagnosis, to selecting a facility, to navigation, to queue management.

The system must provide the following functions in accordance with the other requirements described within this SRS document.

- ➢ Input symptoms to obtain a preliminary diagnosis results
- ➢ Join the queue of nearby healthcare facilities
- ➢ Call ambulance for the corresponding diagnosis result
- ➢ Check information for nearby healthcare facilities
- ➢ Check the route to nearby healthcare facilities
- ➢ Check queue information for their queue
- ➢ Leave queue

2.3 User Classes and Characteristics

All users will be:

- ➢ able to access symptom information
- ➢ able to obtain preliminary diagnosis results from machine learning models
- ➢ able to access webpage explaining corresponding diagnosis results
- ➢ able to call ambulance directly from the webpage
- ➢ able to access information about nearby healthcare facilities
- ➢ able to access information about route to the selected healthcare facility
- ➢ able to register for queue for one nearby healthcare facility

We have further identified three key user classes, which will have added access rights on top of the ones already mentioned above:

1. Super admin/ Singapore Ministry of Health(MOH):
   - ➢ able to access and modify queue information from all registered healthcare facilities
   - ➢ able to access the registered personal particulars for verification
2. Admin user/ Healthcare facilities
   - ➢ able to access and modify queue information from their own queue management system
3. Registered Users
   - ➢ able to leave the queue for the registered queue
   - ➢ able to access their queue information using NRIC

2.4 Operating Environment

Hosting server:

1. Heroku

Operating System

1. Windows 8 OS and above
2. MacOS

NPM dependencies

1. concurrently: 7.0.0,
2. cors: 2.8.5,
3. dotenv: 16.0.0,
4. express: 4.17.3,
5. mongoose: 6.2.3,
6. nodemon: 2.0.15,
7. set-interval-async: 2.0.3
8. chakra-ui/react: 1.8.7,
9. react-google-maps/api: 2.8.1,
10. axios: 0.26.1,
11. react-geocode: 0.2.3,
12. react-hook-form: 7.28.1,
13. react-icons: 4.3.1,
14. react-router-dom: 6.0.0-beta.0,
15. react-select: 5.2.2
16. react-toastify: 8.2.0

2.5 Design and Implementation Constraints

The Personal Data Protection Act (PDPA) sets out the law on data protection in Singapore. The PDPA establishes a data protection law that comprises various rules governing the collection, use, disclosure and care of personal data. Owing to Personal Data Protection Act 2012, the application needs government regulatory authority to approve usage of singpass for verification of users. Also, locations, symptoms and illness can only be stored on client side servers for data privacy reasons.

Our application uses the MongoDB Atlas Free Tier, which only allows 500 concurrent users, a max of 500 collections, and 512MB of storage. This could greatly restrict the scalability of our application in the long run.

2.6 User Documentation

User manual including all possible flows and explanation of user interface is provided in the Appendix E. Users can access hotline support from respective healthcare facilities from the healthcare information page.

2.7 Assumptions and Dependencies

We have identified the following assumptions for our problem formulation, solution effectiveness and implementation results.

Problem assumption

- ➢ Multiple conflicting sources of medical information and anecdote exist online that results in difficult screening for useful information or reliance on incorrect information
- ➢ Limited knowledge to self diagnose
- ➢ Time and cost inefficient to access healthcare facilities whenever symptoms arise

Solution assumption

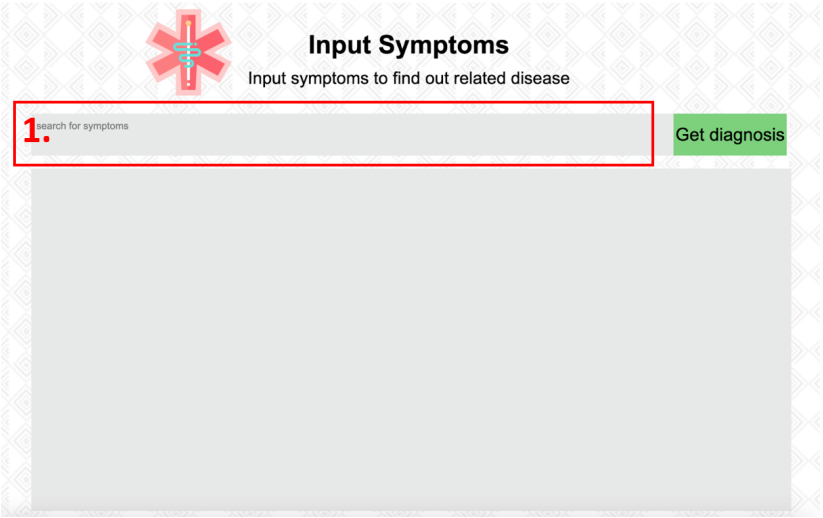
- People would trust and follow advice from a machine learning model

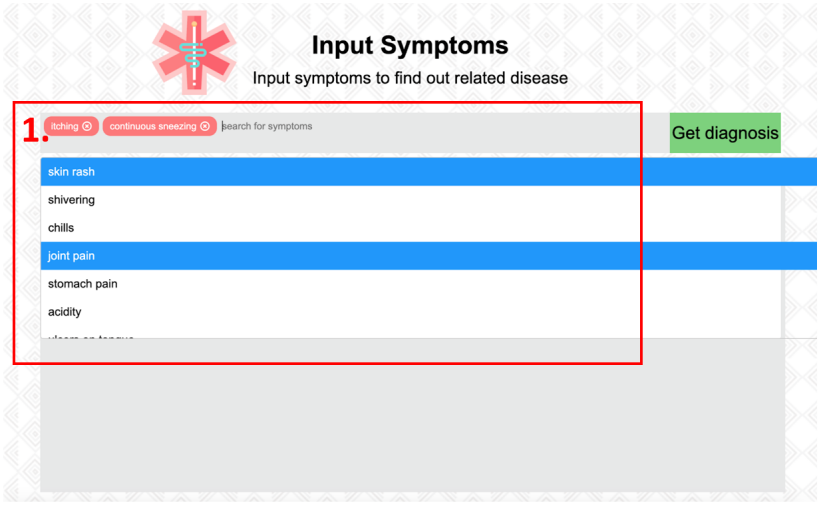
Implementation assumption

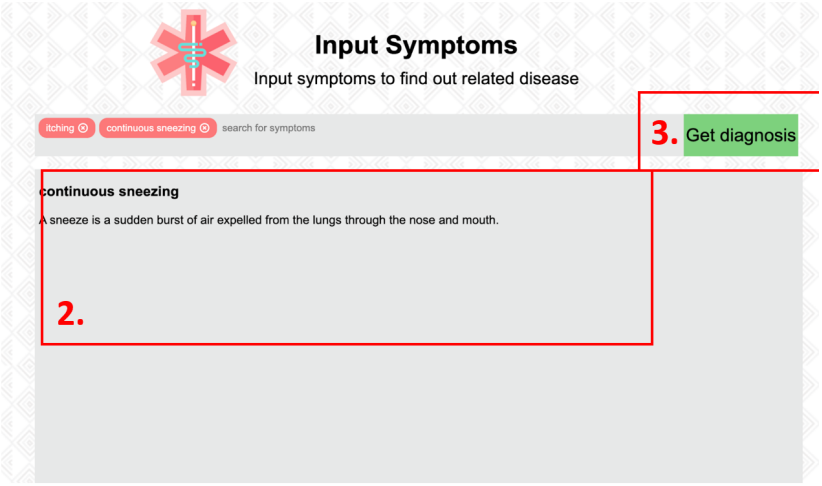
- The model was built in a way that allows for quick access and accurate prediction. So they can trust us with their symptoms

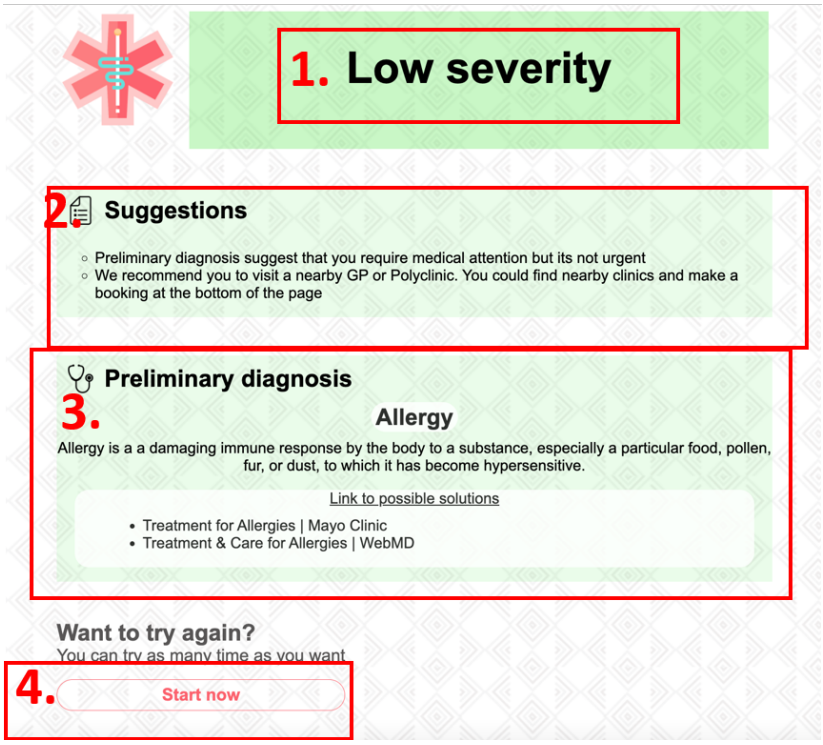
3. External Interface Requirements

3.1 User Interfaces

Interface	Explanation
	**Home page** 1. Guide users to the three respective modules **Diagnosis, Hospital facilities near you and Check queue status.** 2. The icon on the top left will always direct user to the home page in our web application.
	**Symptom input page** 1. The search bar and dropdown menu is to provide users with options of symptoms available in our application. 2. After the user enters each symptom, a description of said symptom will appear in the box below as well to help the user better evaluate if they actually have said symptom. 3. The button **get diagnosis** directs the user to the diagnosis page with their corresponding diagnosis result.


Input Symptoms
Input symptoms to find out related disease
1.
itching
continuous sneezing
search for symptoms
Get diagnosis
skin rash
shivering
chills
joint pain
stomach pain
acidity


Input Symptoms
Input symptoms to find out related disease
itching
continuous sneezing
search for symptoms
3.
Get diagnosis
continuous sneezing
A sneeze is a sudden burst of air expelled from the lungs through the nose and mouth.
2.



Diagnosis page

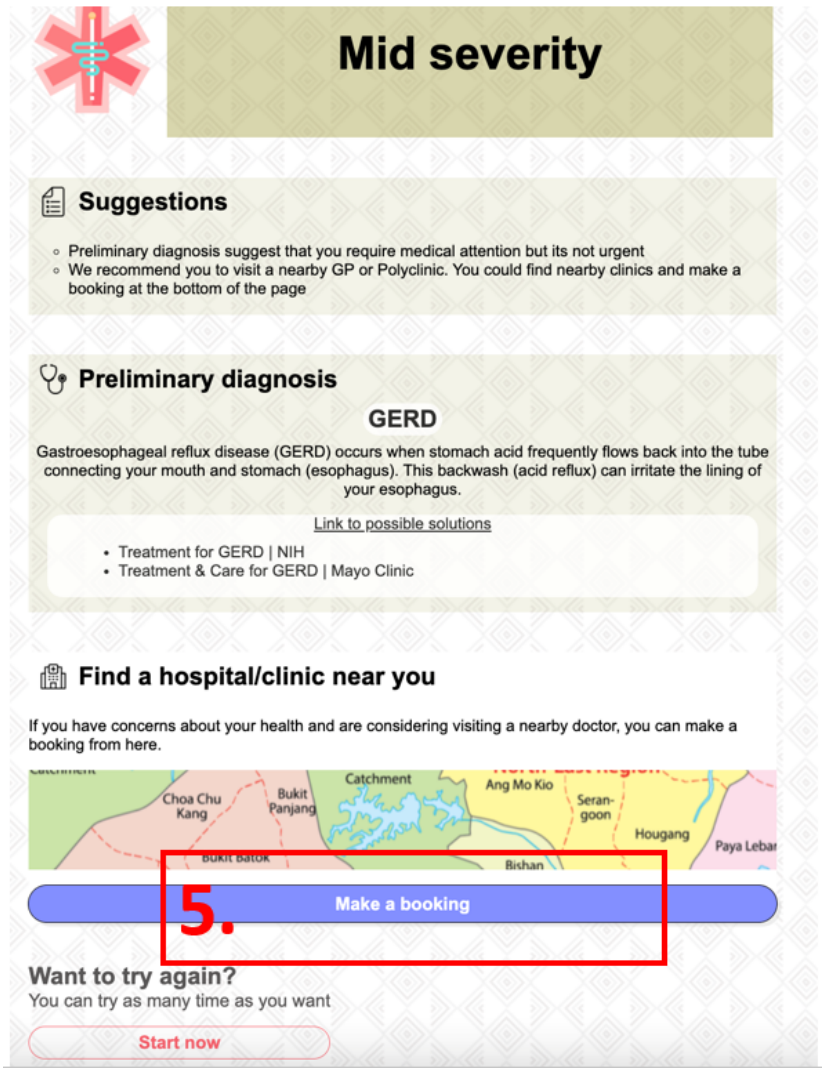
This is an example of the low severity page.

1. Right on top, is the severity of the diagnosis our model has predicted based on the symptoms entered by the user. This will help determine the next course of action.

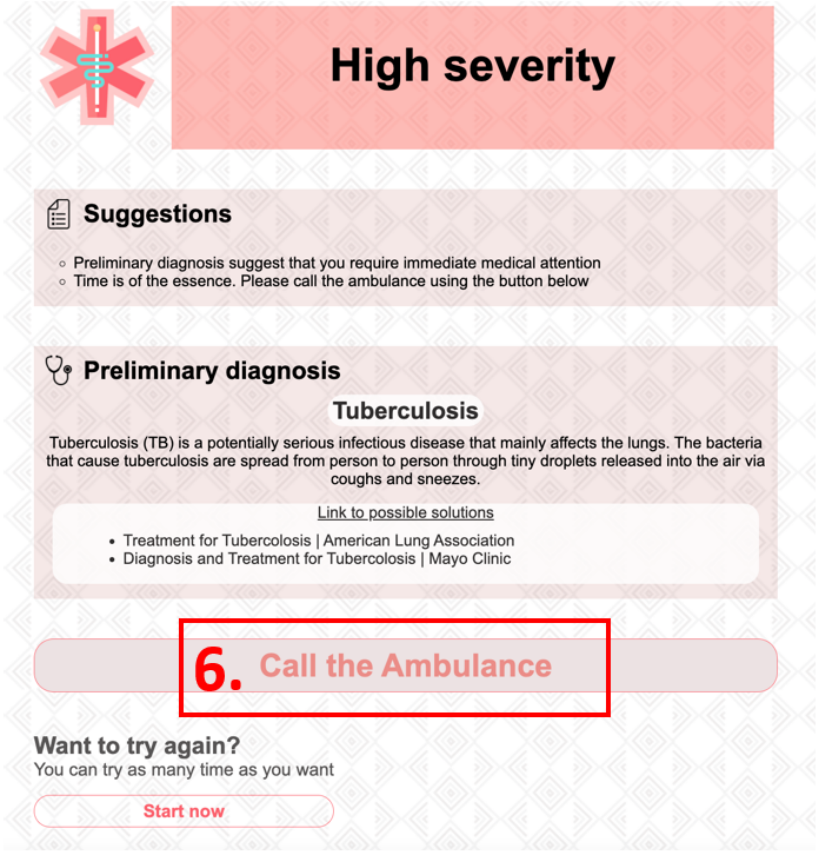
2. Right below, Some suggestions will be provided on what are the actions the user can take next.

3. Below that, a preliminary diagnosis box is provided, with the diagnosis obtained, a short summary of the diagnosis, and a few **links** to self remedies that the user can consider.

4. Finally, there is a **start now** button, which the user can use if they want to try the diagnosis process again.



5. For mid severity diagnosis, the app will next show a find hospital/clinic near you box where the user can select the **make a booking** button to proceed with finding a hospital and getting a queue number.



6. For the high severity diagnosis, there will be a **call the ambulance** button next. This is to help patients with urgent medical needs to access the healthcare facilities as quickly as possible.

Nearby hospital page

1. The user can enter their postal code to find the hospitals nearest to that postal code.

2. The user can also turn on their GPS and click on the current location button to find the hospitals nearest to them.

3. Alternatively, the user can scroll around on the map to manually select the hospital they want.

4. The map can be zoomed in and out as needed.

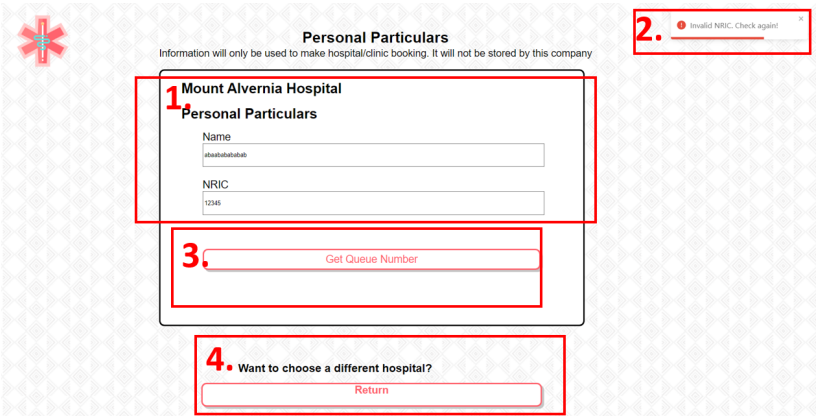
5. Upon clicking a particular hospital, the map will pan to that particular hospital's location.

6. The hospital will be highlighted in black in the menu on the left.

7. There are 2 buttons displayed in the menu, the globe button links to the hospital website, and the button below shows the shortest path from the current location to the hospital that the user can take.

8. A popup in white will be displayed from the hospital's marker on the map, with some relevant details about the hospital such as the contact details, address, opening hours, rating and the number of beds.

9. At the bottom, there is a join queue button which the user can select to join the hospital's queue.



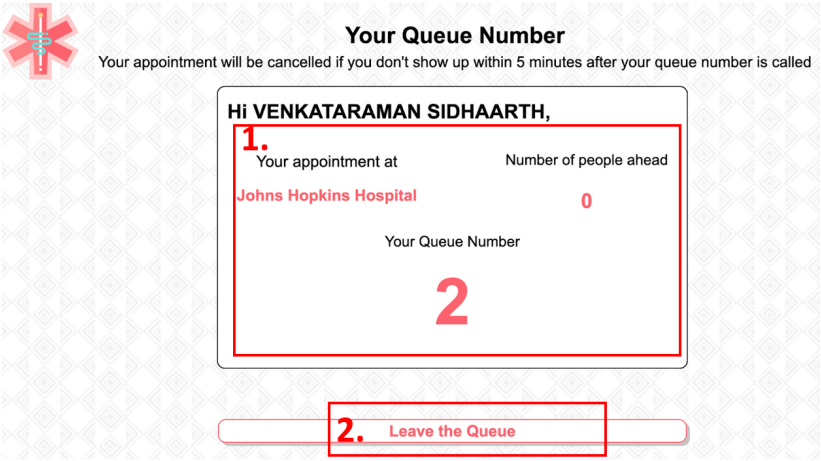
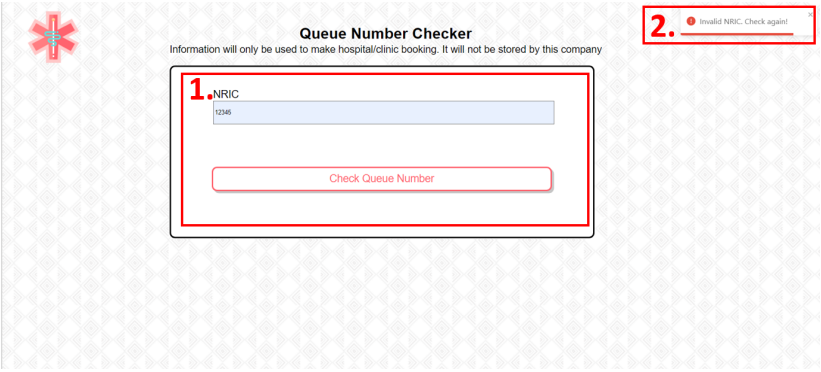
Join queue page

1. The user will need to enter their name and NRIC.

2. If the user enters an invalid NRIC, a popup showing 'invalid NRIC please try again' will be shown.

3. After that, they will have to select **get queue number** button below the input fields to join the queue of the particular hospital.

4. There is a return button available at the very bottom

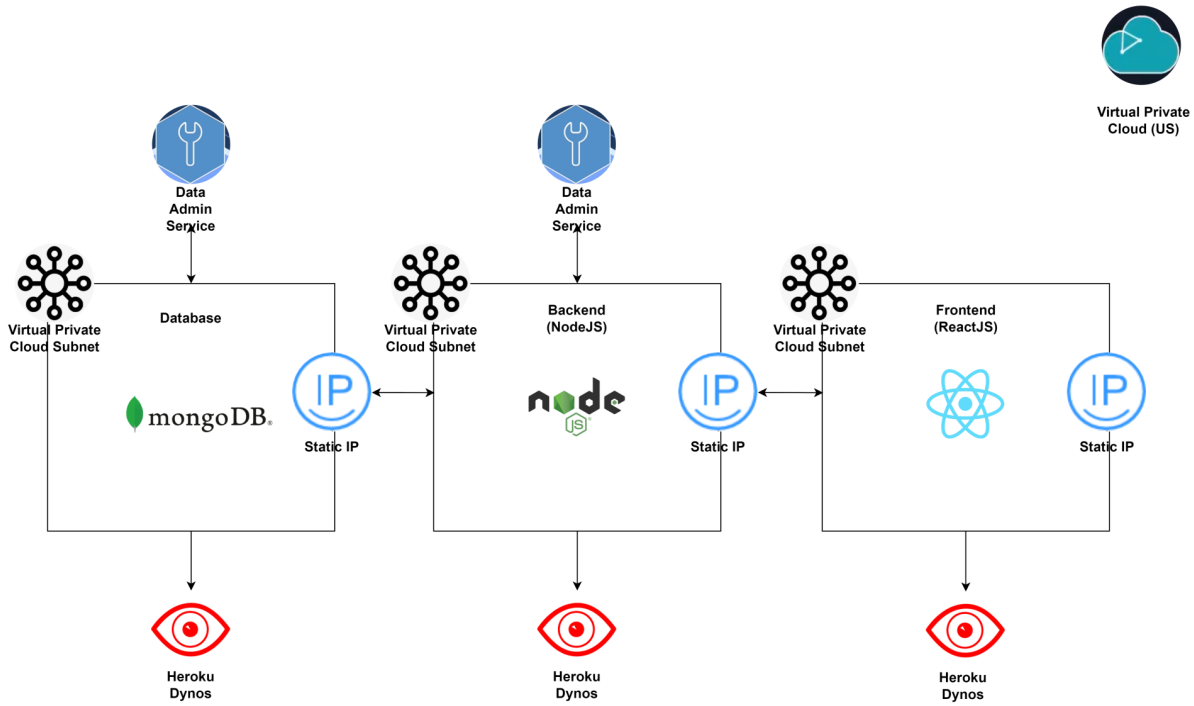
	that the user can select if they decide they want to join a different hospital's queue as well. This button will take them to the nearest healthcare facility page again.
	**Queue information page** 1. The queue details including their name, the hospital, the number of people ahead of them, and their queue number will be shown in this page. 2. A leave the queue button is available at the bottom of the page if the user wishes to leave the queue at any point of time.
	**Check queue page** 1. The user needs to enter their NRIC and select the Check Queue Number button to access the personal particulars page where the user can view their queue details as shown above. 2. The user needs to have already joined a queue and have entered their correct NRIC, or a pop up saying invalid NRIC, check again! Will be shown.

3.2 Hardware Interfaces

For the proper functioning of our web application, all server-side components must execute on server-class computers. All client-side components must execute on workstation-class and personal-class computers.

Client computers must have GPS to access the nearby healthcare facilities.

3.3 Software Interfaces



A combination of technologies is used. The frontend client is coded with Javascript and the React.js framework is used to make interactive user interfaces. The frontend will interact with the backend which is built using Node.js with Express framework to facilitate communication between the application's components. The database is developed using MongoDB for storage of data, in which we employed the Data Admin Service for easy management of databases and an improvement in working efficiency and data security. Heroku Dynos is used to monitor key metrics such as running status and performance of service instances. The entire web application is hosted on Heroku's virtual private cloud(US).

3.4 Communications Interfaces

REST constraints are met with a client-server architecture made up of clients, servers, and resources, with requests managed through Hypertext Transfer Protocol (HTTP).

HTTPS Communication (Web Application to Server API)

- HTTP encrypted by TLS which provides authentication of the website and associated web server with which one is communicating, which protects against man-in-the-middle attacks.
- Additionally, it provides bidirectional encryption of communications between a client and server, which protects against eavesdropping and tampering with or forging the contents of the communication.
- HTTP Post function used mainly for creating user in queue and sending symptoms to the machine learning model

- HTTP Get function to retrieve route and current location for route planning and queue information from database
- HTTP Delete function to remove queue information from database

MongoDB Database Server Connection

- Mongoose will open a active DB connection to the collections to perform Create, Read, Update, Delete (CRUD) operation via HTTP route
- Access the database via MongoDB Compass to debug web app

4. System Features

4.1 Searching for relevant symptoms

4.1.1 Description and Priority

Allows a user to select all the symptoms that are applicable to them
High priority

4.1.2 Stimulus/Response Sequences

1. User clicks on “Diagnose” section of the landing page to arrive at the “Diagnose” web page
2. User types the symptom that he is that he is experience in the free text search field
3. User input in step 2 will query a predefined repository containing all symptoms
4. Matching symptom will be shown in the drop down menu for the user to click
5. The user can then click on it to add it into the list of selected symptoms

4.1.3 Alternative Flows

AF 1: If the symptom that the user searches for is not found
1. Drop down menu shows ‘No options available’

4.1.4 Functional Requirements

REQ-1: The user shall be able to search and select symptoms which are relevant to him
1.1 If a symptom exists in the database, then it will show up in the dropdown menu for the user to select
1.2 If the entered symptom is not in the database, the user display should indicate that no such symptom exist

4.2 Show Symptoms Information

4.2.1 Description and Priority

Shows a short description of the symptom when a user selects it
High priority

4.2.2 Stimulus/Response Sequences

1. User has typed in his symptom into the search box
2. The symptom he entered matches with that which can be found in the symptom list and this a result is returned
3. User clicks on returned result
4. A short textual description of the symptom is shown on the screen

4.2.3 Alternative Flows

- NIL

4.2.4 Functional Requirements

REQ-1: The user shall receive information based on the symptom entered
1.1 The information on the symptom displayed to the user should be in a descriptive text
1.2 If the entered symptom is not in the database, the user display should indicate that no such symptom exist

4.3 Get Diagnosis from Selected Symptoms

4.3.1 Description and Priority

User can gain insights on the likely diagnosis of the medical condition, its corresponding severity and be advised on the next actional step to take
High priority

4.3.2 Stimulus/Response Sequences

1. User has selected all the relevant symptoms that are applicable to him
2. User pressed the 'Get Diagnosis' button
3. System will call rest API of machine learning model to generate a possible diagnosis based on selected symptoms
4. System will then classify the diagnosis into one of three categories: Low, Mid, High and redirects users to page accordingly to display what actionable steps they can take next

4.3.3 Alternative Flows

AF 1: User clicks the 'Get Diagnosis' button without selecting any symptoms
1. A banner will appear, telling him that he is unable to get a diagnosis and to select at least one symptom

4.3.4 Functional Requirements

REQ 1: The user shall receive a diagnosis based on different severities and be advised on remedy steps
1.1 The user shall receive the diagnosis based on 3 severities, low, mid and high
1.1.1 The machine learning model must return a diagnosis based on the input symptoms
1.1.2 The machine learning model must return the diagnosis in the correct classification

1.2 The user shall receive remedy actions based on the different severities of the diagnosis
1.2.1 The user shall receive suggestions and preliminary diagnosis for all classifications of diagnosis
1.2.1 The user shall be given the option to make a booking if the diagnosis is of mid severity

4.4 Healthcare Facilities Near User

4.4.1 Description and Priority

Healthcare Facilities Near User feature is meant to aid the user in deciding which healthcare facility to visit. Users will be able to see a map showing all public healthcare facilities in Singapore and their information. Information such as number of beds, phone number, address, opening hours, website and directions (distance and duration of travel) to the healthcare facility are provided.
High Priority

4.4.2 Stimulus/Response Sequences

1. From the home page, user clicks on “Healthcare facilities near you” card
2. User redirected to the maps page
3. User prompted to allow usage of current location
4. User can click on the marker on the map or a card from the selection pane to view healthcare facilities information
5. If user decides on the facility to go, the user can click the “Join queue” button to be redirected to the page where he can proceed to join the queue of the selected facility

4.4.3 Alternative Flows

- NIL

4.4.4 Functional Requirements

REQ-1: User shall be able to view the locations of all healthcare facilities
- 1.1 Locations of all healthcare facilities will be marked on Google Maps using Maps JavaScript API on load of the webpage
- 1.1 The user shall be able to view all the markers of all healthcare facilities regardless of whether GPS is enabled or not

REQ-2: User shall receive details of all healthcare facilities
- 2.1 User shall receive details such as the name, opening hours, ratings, number of beds, telephone number, address of facility when a marker is clicked
- 2.2 The user shall be redirect to the healthcare facility website when the user clicks on the website icon

REQ-3: User shall know the location of nearby healthcare facilities relative to current location
- 3.1 If user has enabled GPS, Geolocation API will retrieve user’s current location and the webpage will fetch and return a list of healthcare facilities within 7km radius from the user
- 3.2 If user has disabled GPS, the webpage will return a list the full list of all healthcare facilities

REQ-4: User shall know the location of nearby healthcare facilities based on given location
- 4.1 Geocoding API will retrieve the coordinates of the location entered and the webpage will fetch and return a list of healthcare facilities within 7km radius from the given location

REQ-5: User will receive information on route to get to the healthcare facility
- 5.1 When GPS is enabled, using Directions API, webpage will display the recommended route based on current traffic conditions from current location to destination

5.2 Webpage will display the distance and duration of travel from current location to selected healthcare facility
5.3 When GPS is diabled, using Directions API webpage will display the recommended route based on current traffic from centre of Singapore to destination
5.4 Webpage will display the distance and duration of travel from centre of Singapore to selected healthcare facility

4.5 Join Hospital Queue

4.5.1 Description and Priority

Allows a user to join the hospital queue for a chosen hospital
High Priority

4.5.2 Stimulus/Response Sequences

1. User select a hospital they want to go
2. User will be brought to the “Join Hospital Queue” page
3. User input fill in their name and NRIC
4. User click join queue
5. User join queue successfully and was redirected to a page showing the queue information (Hospital name, queue number, number of people ahead of queue)

4.5.3 Alternative Flows

AF1: If name is invalid
1. Error message showing “Invalid name”

AF2: If NRIC is invalid
1. Error message showing “Invalid NRIC”

AF3: If Name is empty
1. Error message showing “Name is required”

AF4: If NRIC is empty
1. Error message showing “NRIC is required”

4.5.4 Functional Requirements

REQ-1: User shall be able to join the queue
1.1 The system will be able to add the user to the specified hospital queue database

REQ-2: User shall be assigned a queue number
2.1 The system will generate a queue number of the user

4.6 Check Queue Status

4.6.1 Description and Priority

Allows a user in the hospital queue to check his current queue status
High Priority

4.6.2 Stimulus/Response Sequences

1. From the homepage, user select “Check queue status”
2. User input his NRIC
3. User selects “Check Queue Number” button
4. User redirected to page to see all his queue information

4.6.3 Alternative Flows

AF1: If NRIC inputted is not in database
1. Error message showing “User not in queue”

AF2: If NRIC is invalid
1. Error message showing “Invalid NRIC. Check again!”

AF3: If NRIC is empty
1. Error message showing “NRIC is required”

4.6.4 Functional Requirements

REQ-1: User shall be able to see the selected healthcare facility queue he is in
1.1 The system will be able to query which hospital queue database the user is in

REQ-2: User shall be able to see his queue number
2.1 The system will be able to retrieve the user queue number from the hospital queue database

REQ-3: User shall be able to see the number of patients ahead of him in the queue
3.1 The system will be able identify the user and calculate the number of patient ahead of him in the hospital queue database

REQ-4: User shall be able to leave the queue
4.1 The user will be able to leave the queue through inputting personal particulars
4.1 The system will delete the personal particulars from the queue of the corresponding hospital

5. Other Nonfunctional Requirements

5.1 Performance Requirements

1. The predictive accuracy of the machine learning models should be greater than 90%
2. When a user submits his symptoms, the machine learning model should return a diagnosis in less than 2s
3. When a user checks the directions to hospital facility, the website should display the route in 3 seconds
4. When a user enables the location feature on the map, the website should display his current location within 5 seconds with an accuracy of 4.9 m (16 ft.) radius under open sky.
5. Whenever a user joins or leaves a queue of a healthcare facility, the queueing system should be updated in less than 10s
6. When a user checks for his queue number, the website should return the information is less than 5s
7. The “Call an Ambulance” function must connect the user directly to the emergency hotline within 30s to immediately receive medical assistance

5.2 Safety Requirements

1. Diagnosis from the machine learning model should be accurate and should not misclassify a user especially for cases of high severity
2. Suggestions for treatments and self-remedies should be aligned with proper healthcare practices
3. Diagnosis and suggestions for treatments should only serve as a guide and should not be treated as proper professional medical advice. Users should be made aware that seeking proper medical advice should still be recommended
4. Directions to healthcare facility should be the shortest route possible based on duration and distance to ensure that treatment is provided to the user as soon as possible
5. Estimation of Duration and Distance should be accurate with a minimal margin of error and adjusted to current traffic conditions to ensure that there is no delay of treatment to the user.

5.3 Security Requirements

1. The queue system can only be joined by individuals with proper authentication (using NRIC)

2. Users' personal information shall be protected within our database
3. Each healthcare facility's queue system is protected and shall not allow external bodies to retrieve any information
4. Webpage that is making a request to use Geolocation to obtain user's current location has to from a secure context
5. User's current location will not be tracked or stored and will only be used at Client-side

5.4 Software Quality Attributes

5.4.1 Accessibility

1. The website can be used by people with minimal hardware capabilities (mobile-friendly, web-friendly)
2. Symptoms and medical conditions should be supplemented with images to aid ease of understanding for the user
3. User can still join a queue of a selected healthcare facility despite GPS being disabled

5.4.2 Availability

1. The website and all its functions should be available at all times
2. The queue management system of the healthcare facilities should be maintained and made available at all times

5.4.3 Extensibility

1. The nearest healthcare facilities module must be designed to allow the addition and removal of healthcare facilities
2. The diagnosis module must allow the addition of new symptoms and medical conditions
3. The queue management system should allow the creation of new databases for new healthcare facilities

5.4.4 Reliability

1. The queuing system must be synchronized between healthcare facilities and the website
2. The "Call an Ambulance" function must always be connected to the nearest emergency medical hotline
3. Links to possible solutions in the diagnosis function must be workable and not broken

4. The system must be able to ensure that an individual can only be allowed to join the queue once for any given healthcare facility

5. The system must be able to keep track of the number of patients in the queue accurately and should always reflect the correct number of patients at any given time

6. When checking the directions and details of a healthcare facility, latest information should be provided

7. The user authentication system should always be accurate

8. The users' information database will be resiliently distributed to prevent loss of information

5.4.5 Supportability

1. Support Personnel shall have administrative access to the database and all features to ensure ease of access to any defects in the operational environment, which can aid understanding in the defects' characteristics, cause, cure and removal.

5.4.6 Usability

1. When the user types into the search bar for a symptom, the partial search shall return searched recommendation

2. When a user clicks on a healthcare facility marker on google maps, an information box with regards to the selected healthcare facility's information will be displayed in less than 1s

3. When a user clicks on a healthcare facility marker on google maps for more than 2s, an information box with regards to the selected healthcare facility's information will be displayed

4. The user shall be provided with suggested symptoms linked to the searched symptom

5. The website shall provide unambiguous descriptions or titles wherever user input is required.

6. The website shall provide features icons that are relevant and reflect the feature's purpose.

5.5 Business Rules

1. The website shall comply with regulation from the various health authorities (e.g. Ministry of Health).

2. Only registered users are able to use any features on the website.

3. Only registered users with a valid NRIC are able to use the join queue feature.

4. Superusers or Admin are able to update the website's back end and database only when necessary.

6. Testing

6.1 Black Box Testing

We made use of cells shading to indicate the type of inputs

- **Grayed** “Test method” cells refer to a single input from the **invalid** equivalence class.
- **Transparent** “Test method” cells refer to a single input from the **valid** equivalence class

1. Queue Management Component

a. Join Queue

Test ID	Test Case	Test Method		Expected Result	Pass/Fail
		NRIC	Name		
1	User should not be able to join queue when valid NRIC and Name is inputted	Tom	S9876543D	User joins queue successfully and redirected to queue information page	Pass
2	User should not be able to join the queue if NRIC already exists in database	Tom	S9876543D	Error prompt “User is already in Queue”	Pass
3	User should not be able to join the queue when name not provided	Empty(“”)	S9988776D	Error prompt “Name is required!”	Pass
4	User should not be able to join queue when invalid name input with special character	Sh@wn	S9988776D	Error prompt “Invalid Name. Check again!”	Pass
5	User should not be able to join queue when invalid name input with numbers	Sh98wn	S9988776D	Error prompt “Invalid Name. Check again!”	Pass
6	Users should not be able to join queue when NRIC is not provided	Tom	Empty(“”)	Error prompt “NRIC is required!”	Pass

	User should not be able to search for information when invalid NRIC with less than 9 alphanumeric characters are provided	Tom	NRIC: “S98”	Error prompt “Invalid NRIC. Check again!”	Pass
7	User should not be able to search for information when invalid NRIC with less than 7 numbers provided	Tom	NRIC: “S987D”	Error prompt “Invalid NRIC. Check again!”	Pass
8	User should not be able to search for information when invalid NRIC with special characters	Tom	NRIC: “S98*7654D”	Error prompt “Invalid NRIC. Check again!”	Pass
9	User should not be able to search for information when invalid NRIC with invalid first character provided	Tom	NRIC: “P9987654D” (P is an invalid first character for Singapore NRIC)	Error prompt “Invalid NRIC. Check again!”	Pass
10	User should not be able to search for information when invalid NRIC with invalid last character provided	Tom	NRIC: “S99876549” (9 is an invalid last character for Singapore NRIC)	Error prompt “Invalid NRIC. Check again!”	Pass

b. Check patient’s queue component

Test ID	Test Case	Test method	Expected Result	Pass/Fail
1	Able to display information of patients who are in queue database and valid NRIC is provided	NRIC: “S9987654D” (NRIC exists in the database)	Redirected to page displaying patient queue information (Name, hospital, queue number and number of people ahead)	Pass

2	Users who are not in the queue database should not be able get queue information even if valid NRIC is provided	NRIC: “S9999999D” (NRIC doesn't exist in the database)	Error prompt “User not in Queue!”	Pass
3	User should not be able to search for information when no NRIC is provided	NRIC: Empty(“”)	Error prompt “NRIC is required!”	Pass
4	User should not be able to search for information when invalid NRIC with less than 9 alphanumeric characters are provided	NRIC: “S98”	Error prompt “Invalid NRIC. Check again!”	Pass
5	User should not be able to search for information when invalid NRIC with less than 7 numbers provided	NRIC: “S987D”	Error prompt “Invalid NRIC. Check again!”	Pass
6	User should not be able to search for information when invalid NRIC with special characters	NRIC: “S98*?@D”	Error prompt “Invalid NRIC. Check again!”	Pass
7	User should not be able to search for information when invalid NRIC with invalid first character provided	NRIC: “P9987654D” (P is an invalid first character for Singapore NRIC)	Error prompt “Invalid NRIC. Check again!”	Pass
8	User should not be able to search for information when invalid NRIC with invalid last character provided	NRIC: “S99876549” (9 is an invalid last character for Singapore NRIC)	Error prompt “Invalid NRIC. Check again!”	Pass

2. Diagnosis Component

Test ID	Test case	Test method	Expected result	Pass / Fail
1	User should be able to navigate to symptom input page from home page	Click on 'Diagnose' card from home page	Website navigates to symptom input page	Pass
2	User should not be able to get a diagnosis if he doesn't key in any symptoms	Click 'Get diagnosis' without selecting any symptoms	A banner notifying them to select at least one symptom will show	Pass
3	Symptom input box should not overflow if too many symptoms are entered	Add as much symptoms as possible such that the width exceeds the width of the symptom input box	Newly inputted symptoms will wrap to the second row if too much symptoms are added	Pass
4	User should be able to view severity low page if his diagnosis of low severity	Select "continuous sneezing" and click "Get diagnosis"	Low severity with diagnosis as 'Allergy'	Pass
5	User should be able to view severity mid page if his diagnosis of mid severity	Select "Loss of appetite", "blurred and distorted vision" and "phlegm" and click 'Get diagnosis'	Mid severity with 'Hepatitis C' as the diagnosis	Pass
6	User should be able to view severity high page if his diagnosis of high severity	Select 'Chest pain', 'dizziness', 'breathlessness' and 'vomiting and click 'get diagnosis'	High severity with 'Heart attack' as the diagnosis	Pass
7	User should be able to navigate back to home page from symptoms input page	Click on logo at top of page	Website will redirect to home page	Pass
8	User should be able to navigate back to home page from severity high page	Click on logo at top of page	Website will redirect to home page	Pass
9	User should be able to	Click on logo at top	Website will redirect to	Pass

	navigate back to home page from severity mid page	of page	home page	
10	User should be able to navigate back to home page from severity low page	Click on logo at top of page	Website will redirect to home page	Pass
11	Resource links on low severity page should redirect to the correct webpage for diagnosis	For 'allergy' example in test case 4, click on one of the links to possible solutions	A new tab showing how to treat allergies will be made	Pass
12	Resource links on mid severity page should redirect to the correct webpage for diagnosis	For 'Hepatitis C' example in test case 5, click on one of the links to possible solutions	A new tab showing how to treat Hepatitis C will be made	Pass
13	Resource links on high severity page should redirect to the correct webpage for diagnosis	For 'Hepatitis C' example in test case 6, click on one of the links to possible solutions	A new tab showing how to treat Heart Attack will be made	Pass
14	User should be able to navigate back to symptoms input page from low severity page to do a retest	Click on the 'Start over button'	Web Page will redirect to symptoms input page	Pass
15	User should be able to navigate back to symptoms input page from mid severity page to do a retest	Click on the 'Start over button'	Web Page will redirect to symptoms input page	Pass
16	User should be able to navigate back to symptoms input page from high severity page to do a retest	Click on the 'Start over button'	Web Page will redirect to symptoms input page	Pass
17	User should be able to navigate to join healthcare facilities page from the mid severity page	Click on "make a booking"	Web page will redirect to the healthcare facilities page	Pass

3. Map Component

Test ID	Test case	Test method	Expected result	Pass / Fail
1	User should be able to navigate to nearby healthcare facilities page from home page	Click on 'Hospital facilities near you' card from home page	Website navigates to nearby healthcare facilities page	Pass
2	User should be able to find nearby hospitals by manually entering a location (Address/Name/Postal Code)	Enters location (Address/Name/Postal Code) and press enter key	Search results container shows all nearest facilities relative to searched location within a 7 km radius	Pass
3	User should not be able to get any specific nearby healthcare facility if he press enter without entering any location	Press enter without entering any location (Location Name/Address/Postal Code)	A banner notifying them to enter a location (Location Name/Address/Postal Code) will show The list of all healthcare facilities without filtering should be shown	Pass
4	User should not be able to get any specific nearby healthcare facility if he press enter without entering a valid location	Press enter without entering a valid location (For eg. "John")	A banner notifying them to enter a valid location (Location Name/Address/Postal Code) will show The list of all healthcare facilities without filtering should be shown	Pass
5	User should be able to view the entire map of Singapore centred in the middle of Singapore upon navigating to nearby healthcare facilities page	Click on 'Hospital facilities near you' card from home page	Website shows the entire map of Singapore centred in the middle of Singapore	Pass
6	User should be able to view all healthcare facilities markers upon navigating to nearby healthcare facilities	Click on 'Hospital facilities near you' card from home page	Website shows all healthcare facilities markers	Pass

	page			
7	User should be able to view all healthcare facilities in the search container upon navigating to nearby healthcare facilities page	Click on 'Hospital facilities near you' card from home page	Website's search shows all healthcare facilities markers	Pass
8	User should be able to check his current location	Click on current location icon in search bar	Map centres on user's current location	Pass
9	User should be able to reset the map and search results to view the entire map of Singapore and all healthcare facilities	Click on reset icon in search bar	Map shows the entire map of Singapore centred in the middle of Singapore and resets the search results to all healthcare facilities	Pass
10	User should be able to find nearby hospitals relative to his current location	Click on current location icon in search bar	Search results container shows all nearest healthcare facilities relative to current location within a 7 km radius	Pass
11	User should be able to navigate back to homepage	Click on HealthLink icon	Web Page will redirect to homepage	Pass
12	Website links on healthcare facilities should redirect to the correct webpage for healthcare facilities information	For 'Mount Alvernia Hospital' example, click on website icon	A new tab showing the selected healthcare facility's website will be made	Pass
13	User should be able to select any healthcare facilities' marker on the map and view its information	For 'Mount Alvernia Hospital' example, click on its marker in the map	An information box for the selected healthcare facility will show up with information such as opening hours, address, contact number, number of available beds etc.	Pass
14	Users should be able to view the distance of his selected healthcare facility from his current location upon getting	Click on the directions icon from selected healthcare facility in search results	Distance in kilometres to the selected healthcare facility from current location will be displayed on the top right of the website	Pass

	the directions	container		
15	User should be able to retrieve directions to his selected healthcare facility from his current location	Click on directions icon from selected healthcare facility in search results container	Two waypoints for origin (current location) and destination (selected healthcare facility) along with the shortest route will be set	Pass
16	Users should be able to view the duration of travel to his selected healthcare facility from his current location upon getting the directions	Click on the directions icon from selected healthcare facility in search results container'	Duration in minutes to the selected healthcare facility from current location will be displayed on the top right of the website	Pass
17	User should be able to navigate to join queue page of his selected healthcare facility	Click on the marker of a selected healthcare facility on the map and click 'Join Queue"	The join queue page of the selected healthcare facility will open up	Pass
18	User should be able to view the map if GPS is disabled	Disable GPS	The healthcare facility page shows up with the map	Pass
19	User should not be able to get his current location if GPS is disabled	Disable GPS and click on current location icon	A banner notifying that GPS is disabled and current location is set to center of Singapore. Map centers to center of Singapore	Pass
20	If GPS is disabled, clicking on get directions button will show the direction to the destination from the center of Singapore	Disable GPS and click on directions icon	A banner notifying that GPS is disabled and current location is set to center of Singapore. Direction route is shown from center of Singapore to selected healthcare facility	Pass
21	User should still be able to get distance and duration to selected hospital if GPS is disabled	Disable GPS and click on directions icon	A banner notifying that GPS is disabled and current location is set to center of Singapore. Distance and duration of travel to selected healthcare facility is calculated from center of Singapore	Pass

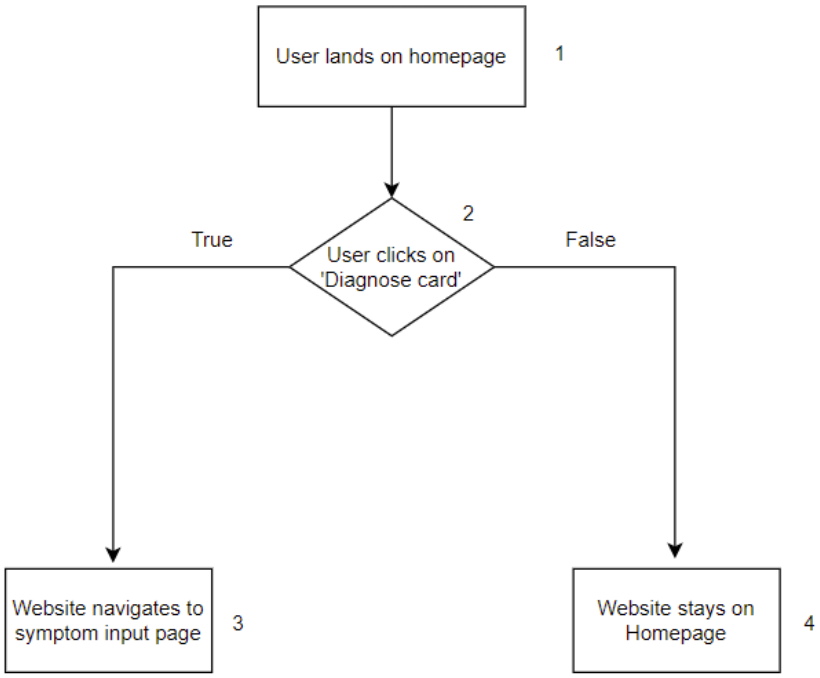
6.2 White Box Testing

The system is segmented into its key components to perform white box testing.

For each component, we first came up with the Control Flow Graph(CFG) based on the code implementation, then we used Basis Graph Coverage (Level 3) to create the test cases. We chose Basis Graph Testing due to its ability to effectively deal with an exponential combination of branches.

6.2.1 Navigating to Symptom input page from Homepage

6.2.1.1 Control Flow graph

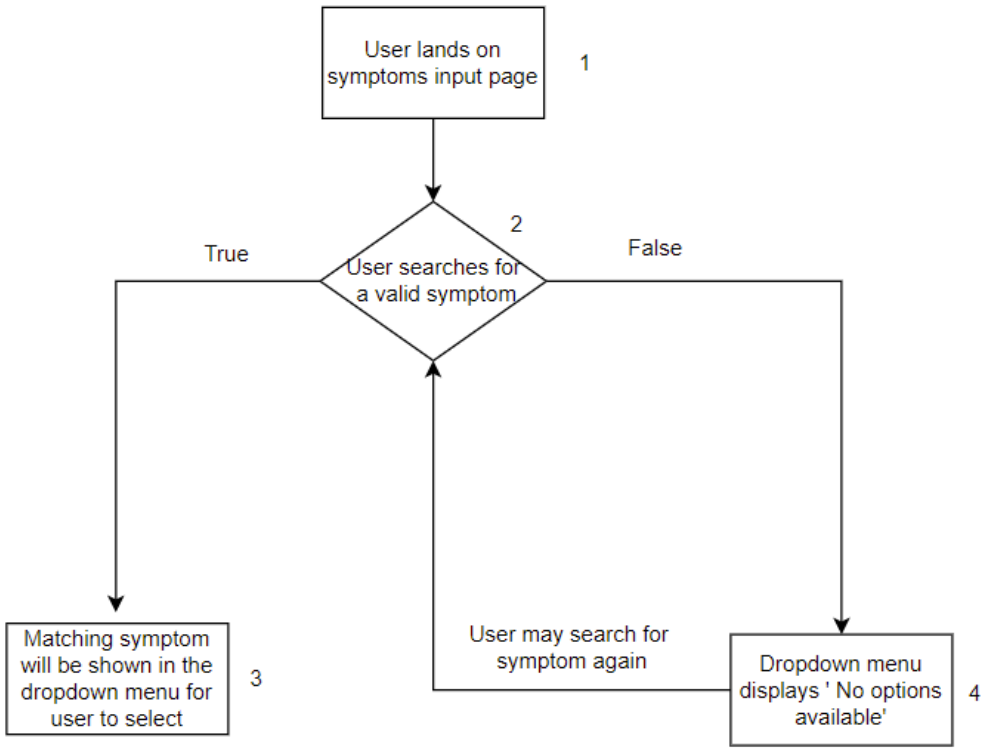


6.2.1.2 Test cases

Test ID	Basis Path	Input test cases	Expected output	Pass/fail
1	1,2,3	User click on ‘Diagnose’ card on homepage	Web page navigates to symptom input page	Pass
2	1,2,4	User doesn’t do anything	Stays on home page	Pass

6.2.2 Navigating to Symptom input page from Homepage

6.2.2.1 Control Flow graph

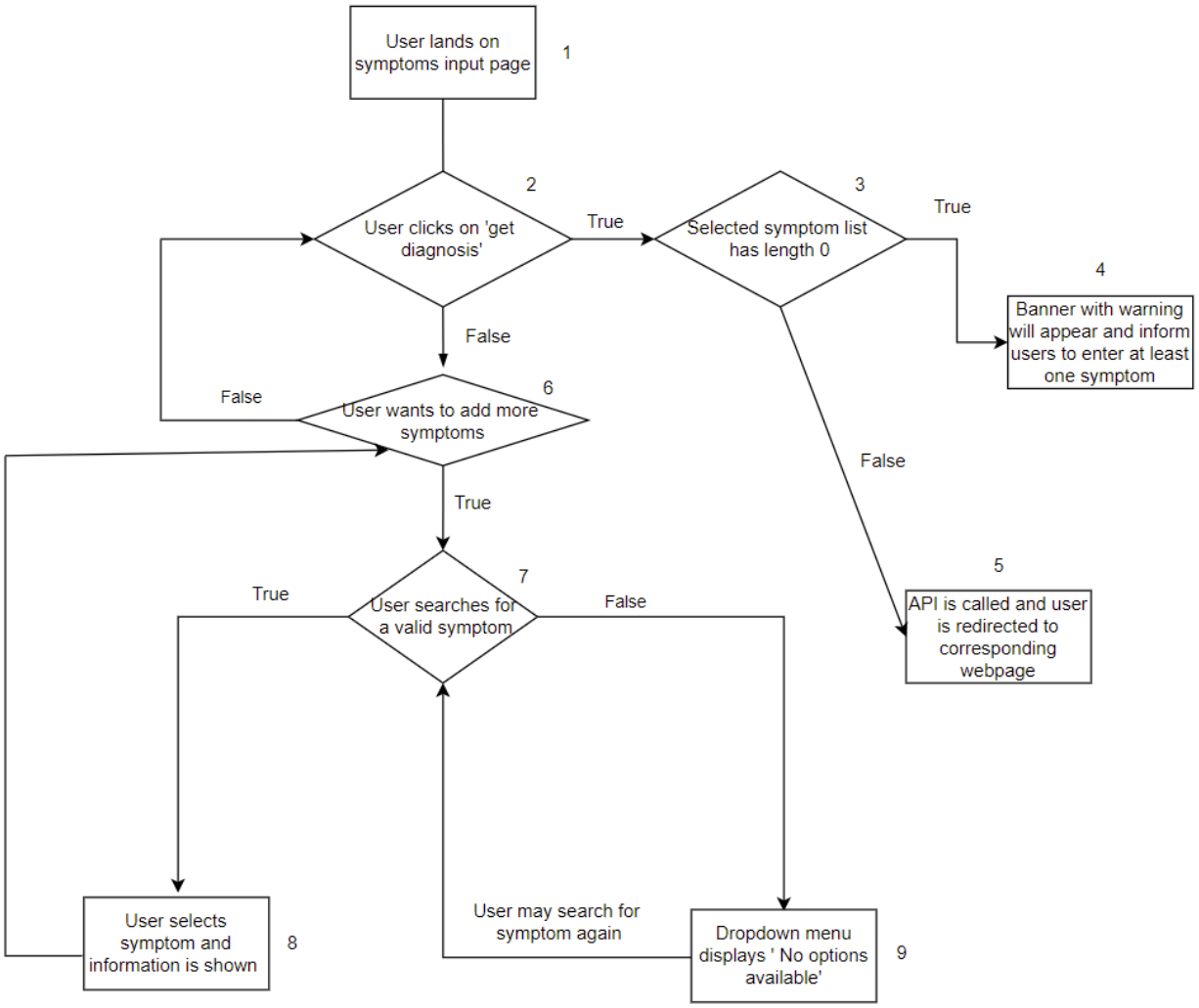


6.2.2.2 Test cases

Test ID	Basis Path	Input test cases	Expected output	Pass/fail
1	1,2,3	Symptom: Chills	Search is valid and chills appears as a symptom to be selected	Pass
2	1,2,4,3	Symptom: Feet pain Re-enter symptoms Chills	When feet pain is entered, ‘No options available’ show. When Chills is re-entered, it shows up as an option to be selected	Pass

6.2.3 Selecting symptoms and getting a diagnosis

6.2.3.1 Control Flow graph



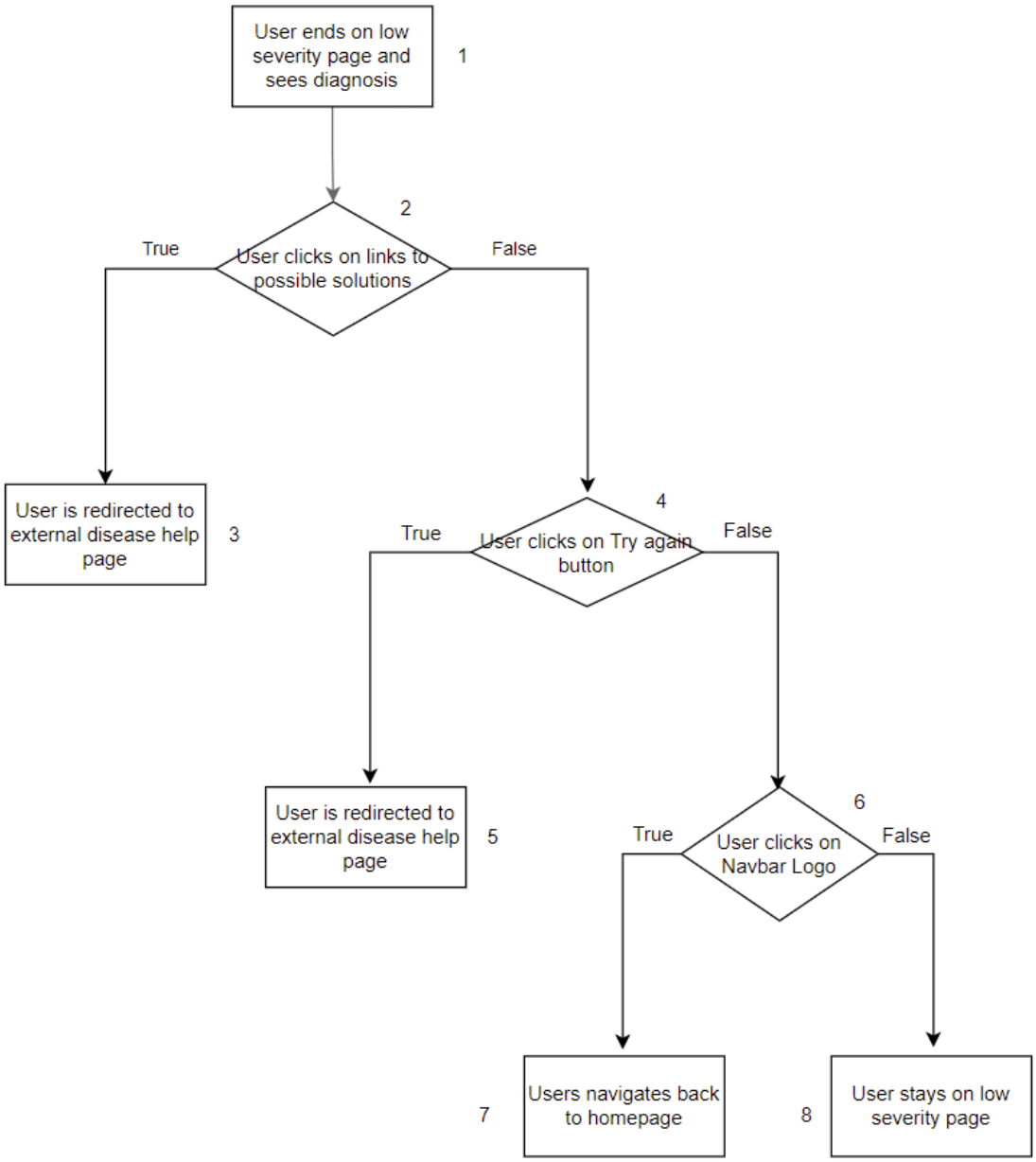
6.2.3.2 Test cases

Test ID	Basis Path	Input test cases	Expected output	Pass/fail
1	1,2,3,4	Click on ‘Get Diagnosis’ button without selecting any symptoms	Banner informing User to ‘Select at least one symptom before clicking on the ‘Get Diagnosis’ button’ should appear at the top right corner	Pass
2	1,2,3,5	Select ‘Chills’ as a	Web page redirects	Pass

		symptom Click ‘Get Diagnosis’ button	to Low Severity page with ‘Allergies’ as the diagnosis	
3	1,2,6,7,8,6,2,3,4	User searches for and selects ‘Chills’ as a symptom User clicks on ‘Get Diagnosis’ button	Description showing ‘The term “chills” refers to a feeling of being cold without an apparent cause. You get this feeling when your muscles repeatedly expand and contract and the vessels in your skin constrict. Chills can occur with a fever and cause shivering or shaking.’ will appear in the grey box below. Web page redirects to Low Severity page with ‘Allergies’ as the diagnosis upon clicking the ‘Get Diagnosis’ button	Pass
4	1,2,6,7,9,8,6,2,3,4	Symptom: Feet pain Re-enter symptom: Chills Click ‘Get Diagnosis’ button	When feet pain is entered, ‘No options available’ show. When Chills is re-entered, it shows up as an option to be selected	Pass

6.2.4 Low severity page

6.2.4.1 Control Flow graph



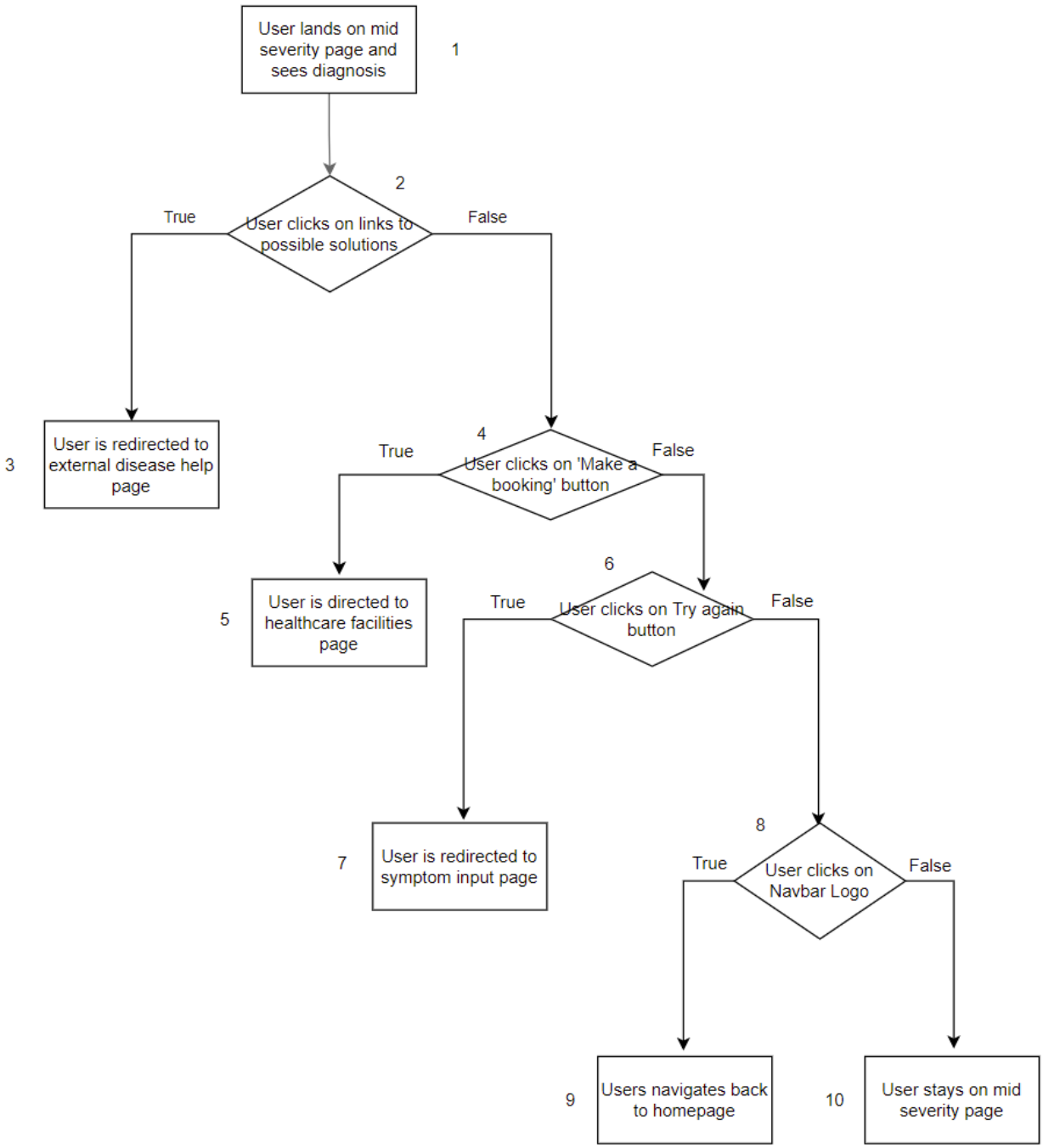
6.2.4.2 Test cases

Test	Basis Path	Input test cases	Expected output	Pass/fail

ID				
1	1,2,3	Symptom: Chills Click Get diagnosis button Click on ‘Treatment for allergies | Mayo Clinic’ hyperlink	New tab pops up with instructions on how to deal with allergies	Pass
2	1,2,4,5	Symptom: Chills Click Get diagnosis button Click on ‘Start Now’ button	Web page redirects to symptom input page	Pass
3	1,2,4,6,7	Symptom: Chills Click Get diagnosis button Click on Navbar Logo	Web page redirects to home page	Pass
4	1,2,4,6,8	Symptom: Chills Click Get diagnosis button Doesn’t select anything on Low severity page	Web page stays the same	Pass

6.2.5 Mid severity page

6.2.5.1 Control Flow graph



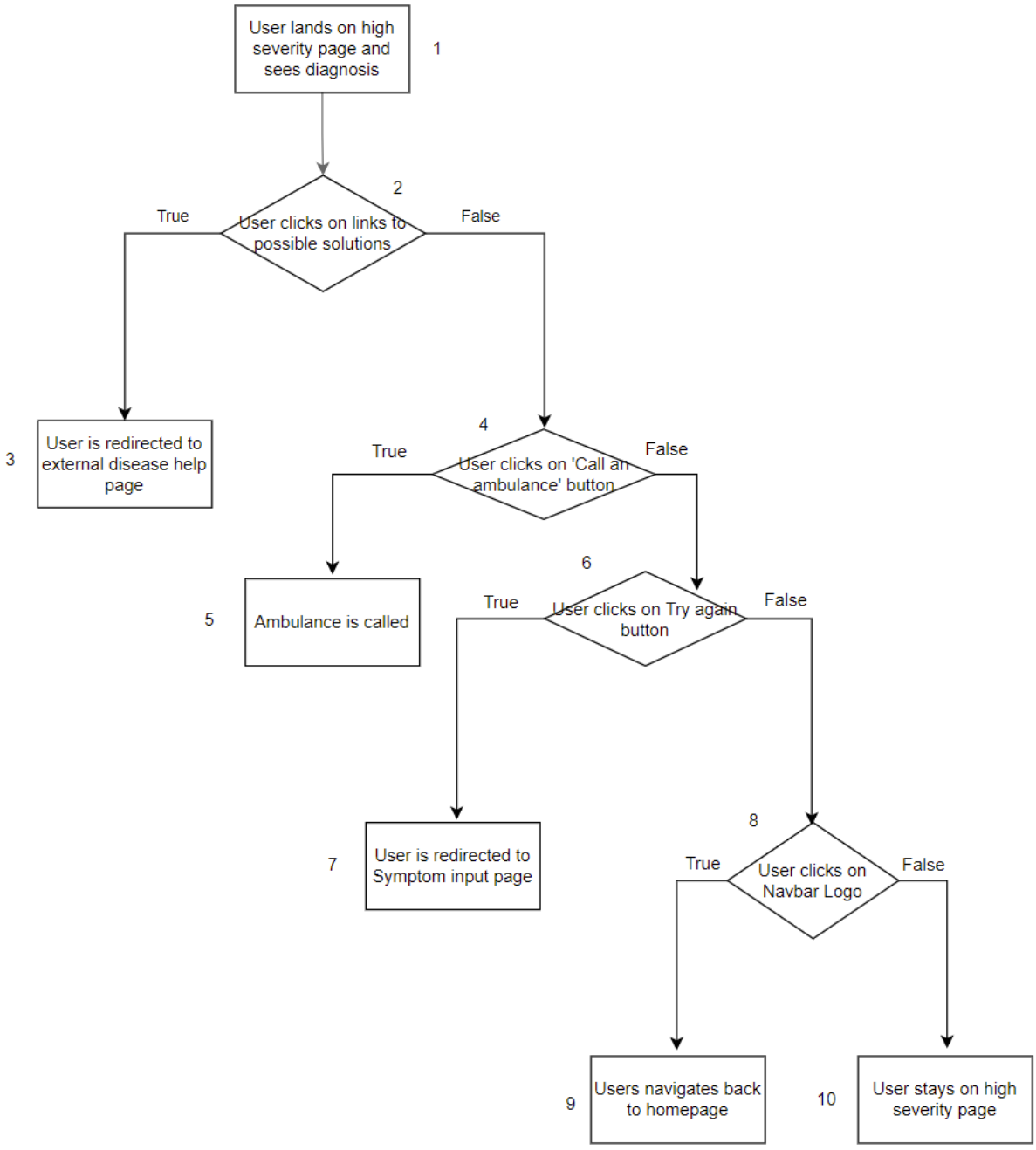
6.2.5.2 Test cases

Test ID	Basis Path	Input test cases	Expected output	Pass/fail
1	1,2,3	Symptom: Rusty Sputum, Distention of abdomen, History of Alcohol Consumption Click Get diagnosis button Click on 'Alcoholic Hepatitis: Causes, Symptoms, and Diagnosis | Healthline'	New tab pops up with instructions on how to deal with Alcoholic Hepatitis	Pass
2	1,2,4,5	Symptom: Rusty Sputum, Distention of abdomen, History of Alcohol Consumption Click Get diagnosis button Click on 'Make a booking'	Web page redirects to healthcare facilities page	Pass
3	1,2,4,6,7	Symptom: Rusty Sputum, Distention of abdomen, History of Alcohol Consumption Click Get diagnosis button Click on Start Now button	Web page redirects to symptom input page	Pass
4	1,2,4,6,8,9	Symptom: Rusty Sputum, Distention of abdomen, History of Alcohol Consumption Click Get diagnosis button Click on Navbar Logo	Website redirects to home page	

5	1,2,4,6,8,10	Symptom: Rusty Sputum, Distention of abdomen, History of Alcohol Consumption Click Get diagnosis button Doesn't select anything on Mid severity page	Web page stays the same	Pass

6.2.6 High severity page

6.2.6.1 Control Flow graph

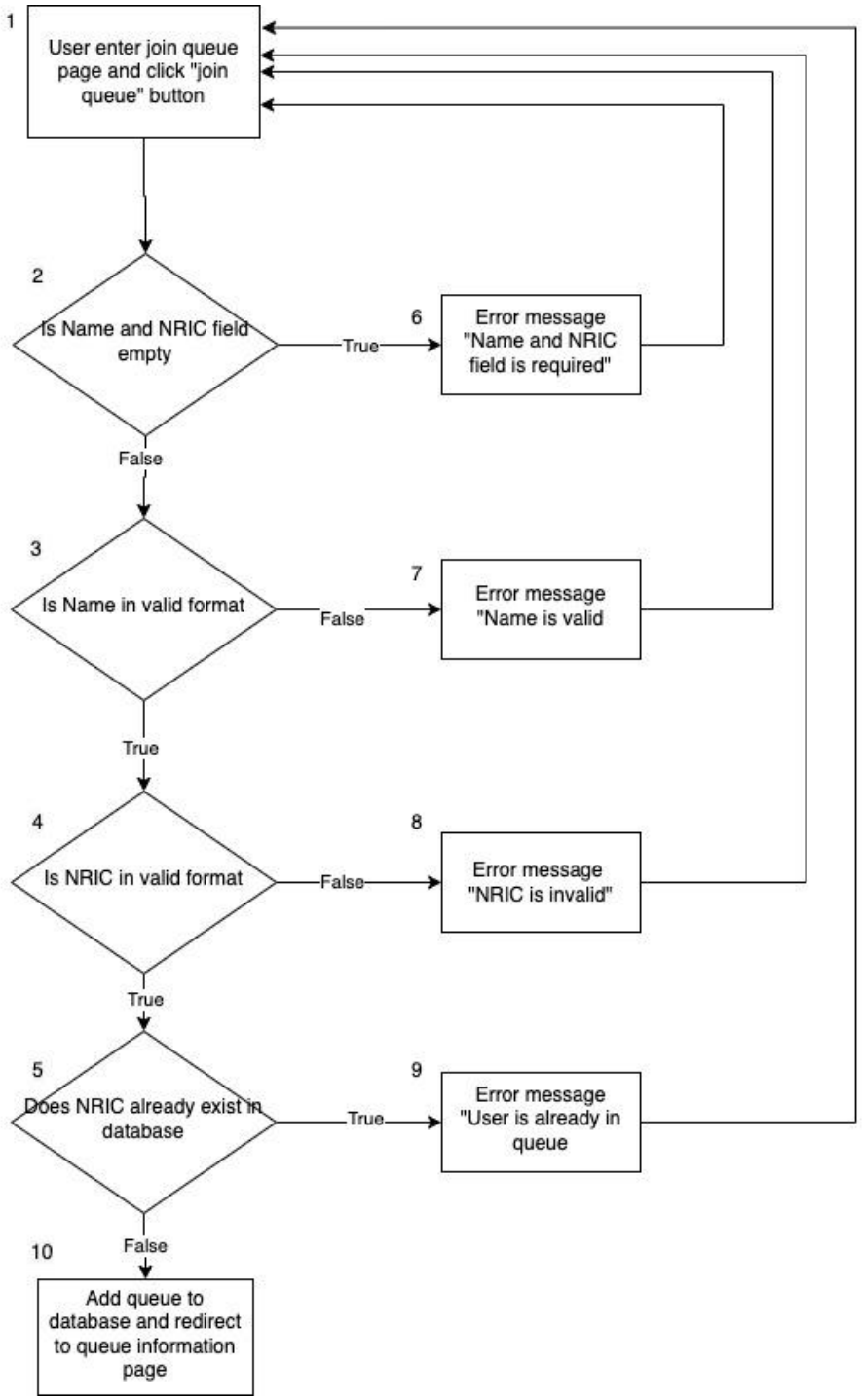


6.2.6.2 Test cases

Test ID	Basis Path	Input test cases	Expected output	Pass/fail
1	1,2,3	Symptom: Chest pain, vomiting, dizziness Click Get diagnosis button Click on 'Heart Attack - Treatment and prevention | WebMD'	New tab pops up with instructions on how to deal with Heart attack	Pass
2	1,2,4,5	Symptom: Chest pain, vomiting, dizziness Click Get diagnosis button Click on 'Call an ambulance' button	Ambulance is called to current location	Pass
3	1,2,4,6,7	Symptom: Chest pain, vomiting, dizziness Click Get diagnosis button Click on 'Try again' button	Web page redirects to symptom input page	Pass
4	1,2,4,6,8,9	Symptom: Chest pain, vomiting, dizziness Click Get diagnosis button Click on Navbar Logo	Website redirects to home page	
5	1,2,4,6,8,10	Symptom: Chest pain, vomiting, dizziness Click Get diagnosis button Doesn't select anything on High severity page	Web page stays the same	Pass

6.2.7 Join Queue

6.2.7.1 Control Flow graph



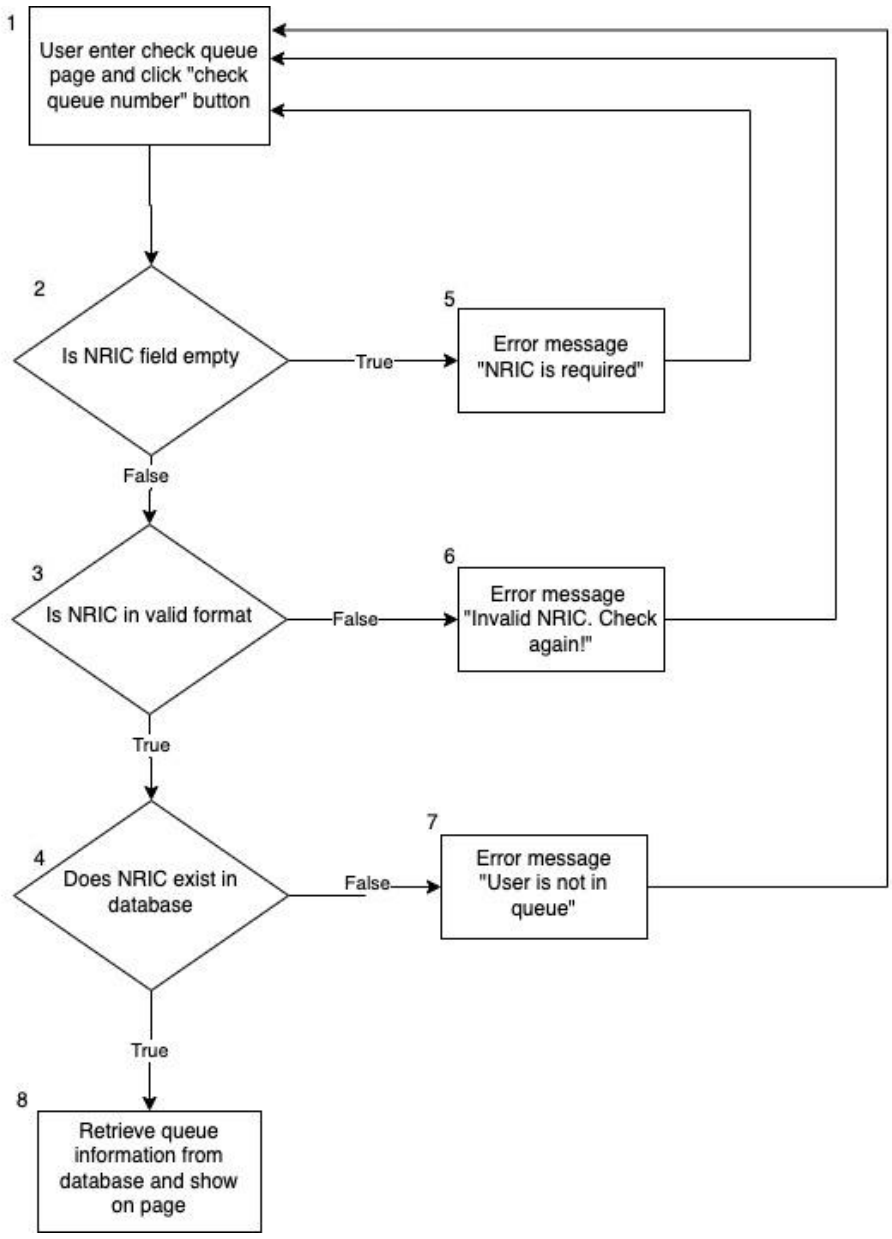
6.2.7.2 Test cases

Test ID	Basis Path	Input test cases	Expected output	Pass/fail
1	1,2,3,4,5,10	Name: Tom NRIC: S9987654D Click submit	Expect the user to be added to database and redirected to information page	Pass
2	1,2,6,1,2,3,4,5,10	Initially leave the name and NRIC field empty, click submit. Then, put name as “Tom” and NRIC as “S9987654D”, click submit	Initially, expect error prompt “Name and NRIC field is required”. Then when input name and NRIC, expect the user to be added to database and redirected to information page	Pass
3	1,2,3,7,1,2,3,4,5, 10	Initially put name as “T@98” and NRIC as “S9987654D”, click submit. Then, put name as “Tom” and NRIC as “S9987654D”, click submit	Initially, expect error prompt “Name is invalid”. Then when input valid name and NRIC, expect the user to be added to database and redirected to information page	Pass
4	1,2,3,4,8,1,2,3,4, 5,10	Initially put name as “Tom” and NRIC as “S99@D”, click submit.. Then, put name as “Tom” and NRIC as “S9987654D”, click submit	Initially, expect error prompt “NRIC is invalid”. Then when input valid name and NRIC, expect the user to be added to database and redirected to information page	Pass
5	1,2,3,4,5,9,1,2,3, 4,5,10	Initially put name as “Tom” and NRIC as “S9987654D” (NRIC is already in database), click submit. Then, put name as “Tom” and NRIC as	Initially, expect error prompt “User is already in queue”. Then when input valid name and NRIC, expect the	Pass

		“S9988776D”, click submit.	user to be added to database and redirected to information page	

6.2.8 Check queue status

6.2.8.1 Control Flow graph

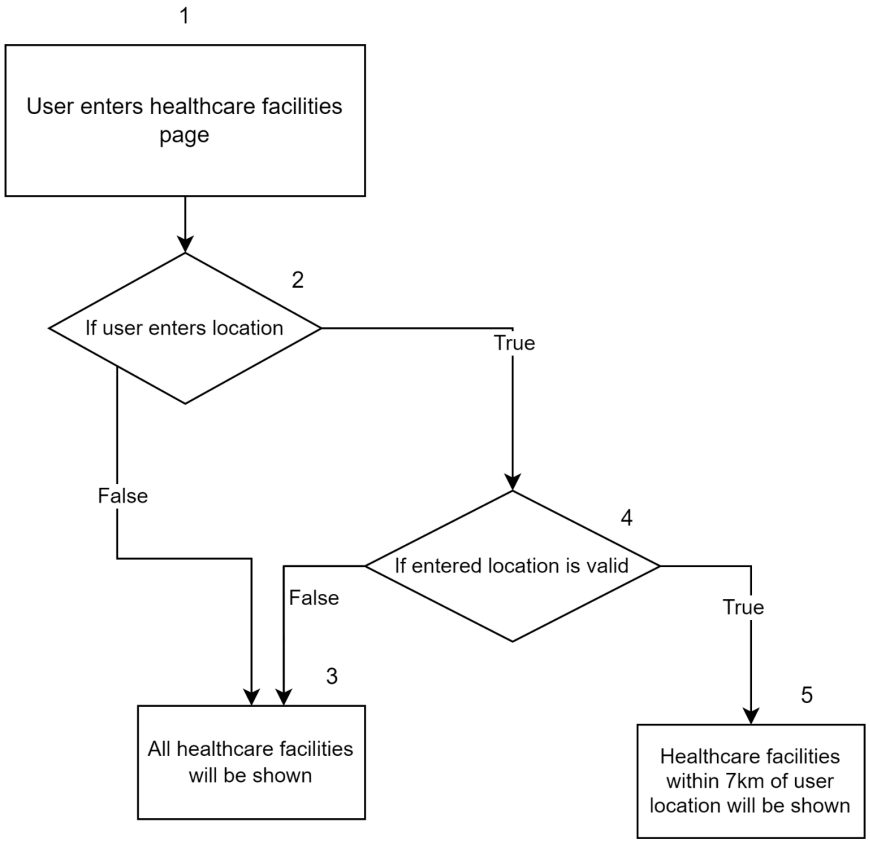


6.2.8.2 Test cases

Test ID	Basis Path	Input test cases	Expected output	Pass/ fail
1	1,2,3,4,8	NRIC: “S9987654D” Click “Check queue button	Expect the screen to display queue information	Pass
2	1,2,5,1,2,3,4,8	Initially leave the NRIC field empty, click submit. Then, put NRIC as “S9987654D”, click submit	Initially, expect error prompt “NRIC is required!”. Then when input valid NRIC, expect the screen to display queue information	Pass
3	1,2,3,6,1,2,3,4,8	Initially put NRIC as “S998@aD”, click submit. Then, put NRIC as “S9987654D”, click submit	Initially, expect error prompt “NRIC is invalid. Check again!”. Then when input valid NRIC, expect the screen to display queue information	Pass
4	1,2,3,4,7,1,2,3,4, 8	Initially put NRIC as “S9999888D” (NRIC is not in the database), click submit. Then, put NRIC as “S9987654D”, click submit	Initially, expect error prompt “User is not in queue”. Then when input valid NRIC, expect the screen to display queue information	Pass

6.2.9 Healthcare Facilities Near User (Entering location)

6.2.9.1 Control Flow graph

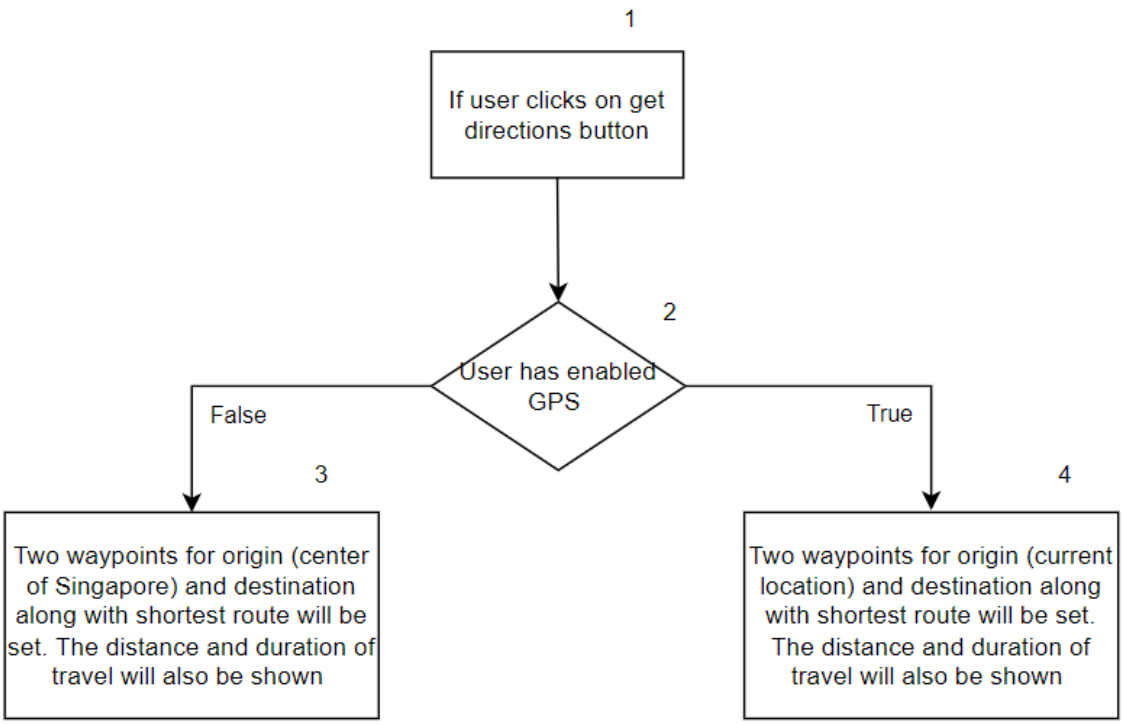


6.2.9.2 Test cases

Test ID	Basis Path	Input test cases	Expected output	Pass/ fail
1	1,2,3	User clicks on “Nearby healthcare facility” page, User leaves search bar empty and press enter key	All healthcare facilities will be shown	Pass
2	1,2,4,3	User clicks on “Nearby healthcare facility” page, User enters location as “Y1sh8n”, location entered is invalid and press enter key	All healthcare facilities will be shown	Pass
3	1,2,4,5	User clicks on “Nearby healthcare facility” page, User enters location as “Yishun”, location entered is valid and press enter key	Healthcare facilities within 7km of location “Yishun” will be shown	Pass

6.2.10 Healthcare Facilities Near User (Clicking on get directions button)

6.2.10.1Control Flow graph

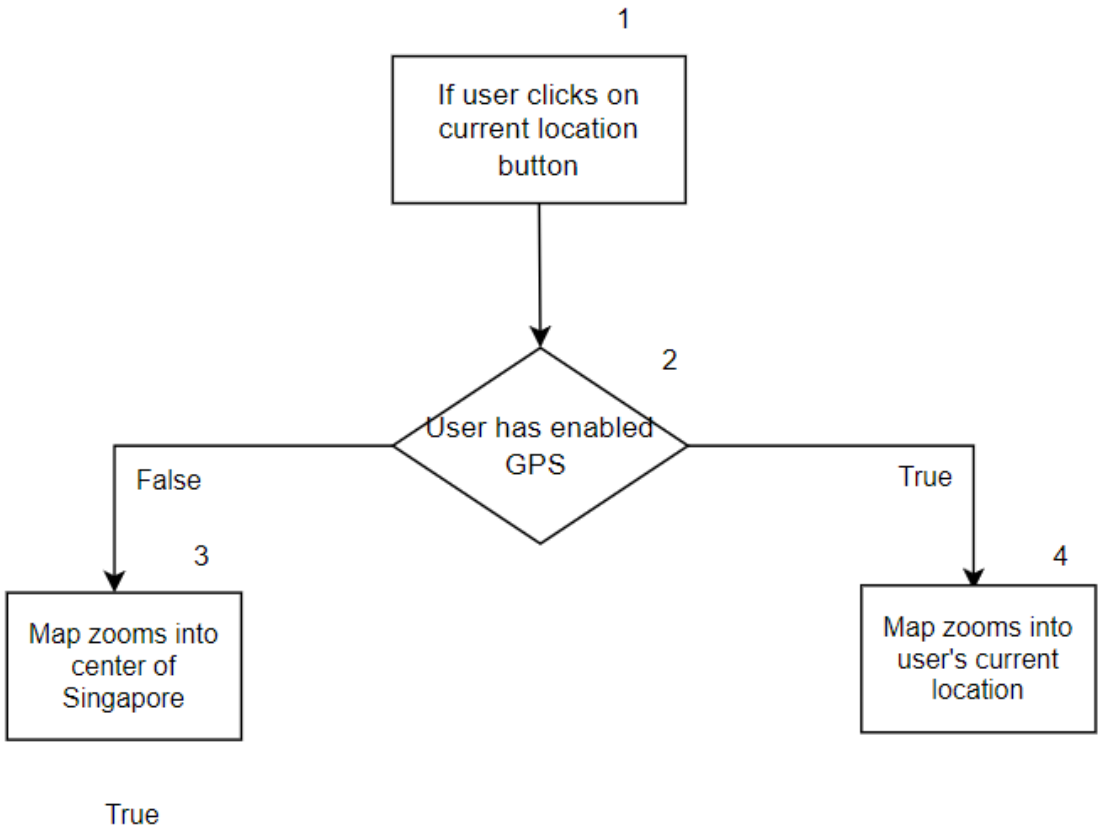


6.2.10.2 Test cases

Test ID	Basis Path	Input test cases	Expected output	Pass/ fail
1	1,2,3	User clicks on get directions button for a selected healthcare facility but has disabled GPS	Two waypoints for origin (center of Singapore) and destination (selected hospital) along with the shortest route will be set. Distance and duration of travel will also be shown	Pass
2	1,2,4	User clicks on get directions button for a selected healthcare facility but has enabled GPS	Two waypoints for origin (user's current location) and destination (selected hospital) along with the shortest route will be set. Distance and duration of travel will also be shown	Pass

6.2.11 Healthcare Facilities Near User (Clicking on current location button)

6.2.11.1 Control Flow graph



6.2.11.2 Test cases

Test ID	Basis Path	Input test cases	Expected output	Pass/ fail
1	1,2,3	User clicks on get current location button but has disabled GPS	Map zooms into the center of Singapore	Pass
2	1,2,4	User clicks on get current location button but has disabled GPS	Map zooms into the user's current location	Pass

Appendix A: Glossary

TERM	DEFINITION
Client	Ministry of Health (MOH)
User	Anybody who uses the web application
Lifecycle	Defined as the time starting from when the User enters the website to the time he leaves it
Patient	A user of the application who chooses to use the application to make an appointment at a healthcare facilities
User information	User's NRIC number and name
Symptom	The surface level problems that the patients identify which will be used as indicators of a potential underlying illness
Diagnosis	A preliminary verdict that is provided to the patient based on the combination of symptoms that the patient has provided
Healthcare Facility	Relevant for medium and high severity, where patients will be referred to healthcare facilities nearby to seek further medical attention. Healthcare facility is a place that provides healthcare. This includes healthcare facilities, clinics, outpatient care centers, and specialized care centers.
Traffic Light Severity	3 severity levels are available and each of them have different implications and follow-up courses of action. Green: Low severity. Links to online remedies provided Yellow: Medium severity. Some online remedies shown. Google Maps API showing nearest healthcare facility and the easiest journey there shown. Booking management System shown to easily book an appointment at the nearest healthcare facility. Red: High Severity. Some online remedies shown. Option to immediately call the ambulance provided as immediate medical attention is deemed to be necessary.

Nearest Healthcare Facility Webpage	A page in the website where the users can see all the nearest healthcare facility with respect to their current location
Healthcare Facility Marker	A google maps marker that when click will open up an infobox that shows information with regards to the healthcare facility
Current location	Current location refers to the user's current location. If GPS has been enabled, current location will refer to the user's current location while is GPS is disabled, current location will refer to the center of Singapore
Get Directions	Get directions will allow the user to get the fastest route from user's location to destination as well as the distance and duration of travel
Join Queue	Join queue refers to the user being able to join the queue of a selected healthcare facility through our application
Queueing System	Functionality where patients can join the queue of their healthcare facility of choice. They are also able to check the number of people ahead of them in the queue.
Personal Particulars	Refers to details regarding the user who wishes to join the queue, consisting of their name and their NRIC.
Queue Number	The number which the user is assigned after joining a queue at a particular nearby healthcare facility. Meant for the user and the healthcare facility to keep track of appointments with patients in an organised manner
Database	Database where each row represents patient's information. It contains user's information, healthcare facility of choice and queue number in the healthcare facility
Machine Learning Model	A machine learning model which outputs a likely diagnosis based on the inputted symptoms. Various techniques have been employed such as Classification and Regression Tree (CART), Random Forest and Logistic Regression
Repository	Refers to a stored location of data and information which can be queried from. The healthcare facilities repository stores information

	such as number of beds, ratings, address, contact number and the symptom repository stores all possible symptoms the User can select
Authentication	Refers to to the process of verifying the authenticity of the user joining the queue
Selection Pane	Refers to a sidebar on the website that allows the user to navigate and select through a list of symptoms or healthcare facilities.
GPS	Global Positioning System (GPS) is a satellite constellation supporting highly accurate positioning, navigation and timing (PNT) measurements worldwide. This aids in the process of tracking the user’s current location to obtain the nearest healthcare facilities.

Appendix B: Analysis Models

Use Case Descriptions

Use Case ID	1		
Use Case Name	Search symptoms		
Created by	Shein Htet	Last Updated by	Shein Htet
Date Created	5th February 2022	Date Last Updated	11th April 2022

Actor:	User
Description:	As a user, I want to be able to enter and select all the symptoms I am experiencing so that I can get an accurate and reliable diagnosis of the medical condition
Precondition:	1. Repository of symptoms has be created and stored
Postcondition:	-
Priority:	High
Frequency of Use:	4 times per lifecycle (assuming that one User has an average of 4 symptoms)
Flow of Events:	1. User clicks on the “Diagnose” section of the landing page to arrive at the “Diagnose” web page 2. User types the symptom that he is experiencing in the free text search field 3. Aforementioned repository will be queried to check if there are any matching entries 4. Matching symptom will be shown in the dropdown menu for the user to click 5. Dropdown menu shows the symptom corresponding to what the user has typed and the user can then click on it
Alternative Flow:	AF-4: If the user is unable to find a symptom 1. The system displays the message “Symptom not found!” for 2 seconds. 2. The system returns to step 2.
Exceptions:	-

Includes:	
Special Requirements:	-
Assumptions:	1. Patient is able to accurately identify the symptom that he is experiencing
Note and Issues:	-

Use Case ID	2		
Use Case Name	Diagnosis		
Created by	Shawn Yap	Last Updated by	Shawn Yap
Date Created	4th February 2022	Date Last Updated	11th April 2022

Actor:	System
Description:	As a user, I want to gain insights on the likely diagnosis of the medical condition, its corresponding severity and be advised on the next actionable steps to take
Precondition:	Machine Learning model is connected to the front end via API
Postcondition:	-
Priority:	High
Frequency of Use:	1-2 times per lifecycle (After getting a diagnosis, the user might want to get a diagnosis again just to be sure)
Flow of Events:	1. User have input the symptoms that he has 2. User pressed “Get diagnosis” 3. System will use the machine learning model to generate a possible diagnosis based on the user symptoms 4. System then display the diagnosis on the webpage for the user as well as what actionable steps to take next
Alternative Flow:	-
Exceptions:	-

Includes:	-
Special Requirements:	-
Assumptions:	1. Machine learning model able to generate diagnosis for all possible symptoms combinations
Note and Issues:	-

Use Case ID	3		
Use Case Name	Show advice for self remedy		
Created by	Shawn Yap	Last Updated by	Shawn Yap
Date Created	4th February 2022	Date Last Updated	11th April 2022

Actor:	System
Description:	As a user, with a low/ mid/ high severity post diagnosis, I will like to receive targeted advice for self remedy
Precondition:	Google Search API connected to the system
Postcondition:	-
Priority:	High
Frequency of Use:	1-2 times per lifecycle
Flow of Events:	1. User have input the symptoms that he has 2. User pressed “Get diagnosis” 3. System has generated a possible diagnosis based on the provided symptoms using the machine learning model 4. System will use the google API to get possible self remedy for the diagnosis generated 5. Website to display the self remedy for the diagnosis
Alternative Flow:	-
Exceptions:	-
Includes:	-

Special Requirements:	-
Assumptions:	1. Google Search API able to generate self remedy for all possible diagnosis
Note and Issues:	-

Use Case ID	4		
Use Case Name	Call the Ambulance		
Created by	V.Sidhaarth	Last Updated by	V.Sidhaarth
Date Created	4th February 2022	Date Last Updated	11th April 2022

Actor:	User
Description:	The website shall provide the user with the option to call the ambulance if they are diagnosed with high severity
Precondition:	1. The system is connected to the call feature of the device
Postcondition:	-
Priority:	High
Frequency of Use:	1 time per lifetime (If the user gets high severity and calls the ambulance, they will not go through the process again most likely)
Flow of Events:	1. The user goes through the diagnosis process, and receives a high severity diagnosis 2. The page displays a call the ambulance button below the diagnosis details 3. The user selects the call the ambulance button 4. The system calls the stored ambulance number through the user's phone
Alternative Flow:	-
Exceptions:	EX1: If the user does not wish to call the ambulance 1. The system will do nothing.

Includes:	-
Special Requirements:	1. User must receive a high severity diagnosis to be able to call an ambulance through the website
Assumptions:	1. User is using their handphone as the primary device to access our website so we can directly use the handphone's calling capabilities to call the ambulance
Note and Issues:	-

Use Case ID	5		
Use Case Name	Show nearby healthcare facilities		
Created by	Ivan Lua	Last Updated by	Ivan Lua
Date Created	5th February 2022	Date Last Updated	10th April 2022

Actor:	User
Description:	As a user, I want to be able to enter my address or enable my GPS tracker to get the nearest healthcare facility relative to my current position
Precondition:	1. User has enabled GPS tracking or manually entered his current address to get the nearest healthcare facilities 2. GoogleMaps API must be running and connected to the website
Postcondition:	-
Priority:	High
Frequency of Use:	1 time per lifecycle (User will want to display nearest healthcare facilities once before selecting a healthcare facility)
Flow of Events:	1. Repository of healthcare facilities will be created 2. Map will be shown using GoogleMaps API 3. User clicks on the "Nearest Healthcare Facility" section of the landing page to arrive at the "Nearest Healthcare Facility" web page

	4. User will manually enter address or enable automatic GPS tracking of his location 5. Healthcare facilities within a radius of 8 kilometres of user's location will be shown on map
Alternative Flow:	AF-S5: If there are no healthcare facilities near to user's current location 1. There will be no healthcare facilities displayed on the map 2. Return to step 4
Exceptions:	EX1: GPS not enabled on user's device 1. System displays a message saying "location not found will be shown. Please enable GPS" 2. The healthcare facilities within a radius of 8 km of the centre of Singapore will be shown
Includes:	-
Special Requirements:	GPS hardware capability must be available in user's device to enable automatic GPS tracking of his location
Assumptions:	1. GoogleMaps API is synchronised with the repository of healthcare facilities such that the location of the healthcare facility is the same.
Note and Issues:	1. Repository of healthcare facilities likely has to be created manually 2. Repository of healthcare facilities has to be updated on a regular basis

Use Case ID	6		
Use Case Name	Show healthcare facility information		
Created by	Ivan Lua	Last Updated by	Ivan Lua
Date Created	5th February 2022	Date Last Updated	11th April 2022

Actor:	User
Description:	When a user selects a healthcare facility, he is likely to want details of his chosen healthcare facility

Precondition:	1. User has enabled GPS tracking or manually entered his address to get the nearest healthcare facilities 2. Google Places API and GoogleMaps API must be running and connected to the website
Postcondition:	-
Priority:	High
Frequency of Use:	1-5 times per lifecycle (User may want to refer to healthcare facility's information for several times to obtain various information)
Flow of Events:	1. User types address into search box or enable GPS and there are results of healthcare facilities within his radius of 8 kilometers 2. User hovers over a selected healthcare facility marker/ clicks on the healthcare facility name on the selection pane 3. Selected healthcare facility's information such as number of beds, opening hours, healthcare facility description, hotline, address etc is shown
Alternative Flow:	-
Exceptions:	EX1: Selected healthcare facility does not have any available information or have missing information 1. The system will do nothing.
Includes:	-
Special Requirements:	GPS hardware capability must be available in user's device to enable automatic GPS tracking of his location
Assumptions:	1. There are healthcare facilities within a radius of 8 kilometres of the user's location. 2. The selected healthcare facility has uploaded their information to Google Places API
Note and Issues:	-

Use Case ID	7

Use Case Name	Join Queue		
Created by	Sng Yi Xuan	Last Updated by	Sng Yi Xuan
Date Created	5th February 2022	Date Last Updated	11th April 2022

Actor:	User
Description:	As a user, I will like to be able to join the queue of a selected healthcare facility
Precondition:	Web page is connected to a database to store information of users that join the queue
Postcondition:	User's information will be updated into the queuing system database
Priority:	High
Frequency of Use:	1 time per lifecycle (User is only allowed to join the queue once for any given healthcare facility)
Flow of Events:	1. User hovers over a selected healthcare facility marker/ clicks on the healthcare facility name on the selection pane 2. User pressed on the "Join Queue" button for the selected healthcare facility 3. User will be directed to the authentication page where he will be required to enter his particulars
Alternative Flow:	-
Exceptions:	-
Includes:	Input particulars
Special Requirements:	-
Assumptions:	All healthcare facilities in Singapore have their physical queues integrated with the online queueing system
Note and Issues:	-

Use Case ID	8		
Use Case Name	Input Particulars		
Created by	Sng Yi Xuan	Last Updated by	Sng Yi Xuan
Date Created	5th February 2022	Date Last Updated	11th April 2022

Actor:	System
Description:	When a user joins a queue for a healthcare facility, the user will be directed to the authentication page where he will be required to enter his particulars for verification
Precondition:	1. System is connected to the databases
Postcondition:	User's information will be updated into the queuing system database
Priority:	High
Frequency of Use:	1-2 times per lifecycle (User might enter the particulars incorrectly the first time round)
Flow of Events:	1. User pressed on the "Join Queue" button for the selected healthcare facility 2. User directed to the authentication page where he will be required to enter his particulars 3. User enters his "Name" and "NRIC" and clicks on "Get queue number" 4. System will add the user's particulars to the database 5. If User is validated, System will extract the user's data and update the queue system database for the respective healthcare facility 6. User is directed to the queue status page and will be issued a queue number
Alternative Flow:	AF-S5: If User is not validated 1. System will display the message "Incorrect name or NRIC. Please try again" 2. Return to Step 3
Exceptions:	EX1: User's particulars already exist in the database, meaning they are already part of a queue 1. System will display the message "User is already part of

	a queue!”
Includes:	Show queue information
Special Requirements:	
Assumptions:	1. Database that stores all users’ personal information exist 2. System has access to the database that stores all users’ personal information
Note and Issues:	There are 2 databases in action here: 1. Database storing all users’ personal information 2. Database to manage the respective healthcare facility’s queue

Use Case ID	9		
Use Case Name	Check queue status		
Created by	V.Sidhaarth	Last Updated by	V.Sidhaarth
Date Created	4th February 2022	Date Last Updated	11th April 2022

Actor:	User
Description:	The website has a check queue status option on its main landing page, which the user is able to select.
Precondition:	1. Their queue status is in the database
Postcondition:	-
Priority:	High
Frequency of Use:	1-5 times per lifetime (an estimation, the user will probably check their queue status at least a few times before their turn comes)
Flow of Events:	1. The user is on the landing page, and selects the third option, check queue status 2. The user is brought to the queue number checker page, where they will be prompted to enter their personal particulars 3. System searches the database to find user's data 4. If user's data is found, user will be redirected to a page displaying all the user's queue information
Alternative Flow:	AF-S2: If the user, particulars and queue details do not exist in the database (meaning the user has not joined any queue yet or they entered the wrong NRIC) but the user still tries to check their queue information. 1. The system displays the message, "Incorrect particulars!". 2. The system returns to step 2.
Exceptions:	-
Includes:	-
Special Requirements:	-

Assumptions:	User has booked a queue number from a healthcare facility using web page
Note and Issues:	-

Use Case ID	10		
Use Case Name	Show queue information		
Created by	V.Sidhaarth	Last Updated by	V.Sidhaarth
Date Created	4th February 2022	Date Last Updated	11th April 2022

Actor:	User
Description:	The user is shown their respective queue details.
Precondition:	1. The user's data is in the database
Postcondition:	-
Priority:	High
Frequency of Use:	1 - 5 times per lifecycle (an estimation, the user will probably check their queue status at least a few times before their turn comes)
Flow of Events:	1. The user has selected get queue number or check queue number 2. System retrieves the queue information from queue management database 3. The user is brought to the next page displaying their queue information. 4. The queue information displayed are the user's name, the healthcare facility, number of people ahead in the queue, and the user's queue number
Alternative Flow:	AF-S4: If the user, particulars and queue details do not exist in the database (meaning the user has not joined any queue yet or they entered the wrong NRIC) but the user still tries to check their queue information. 1. The system returns to step 2.
Exceptions:	

Includes:	-
Special Requirements:	1. System must always be connected to the healthcare facility to be able to display up to date queue information of the healthcare facility whenever the user requests for it
Assumptions:	1. User must have already joined a queue to be able to see their queue information
Note and Issues:	-

Use Case ID	11		
Use Case Name	Leave the queue		
Created by	Zhou Qiren	Last Updated by	Zhou Qiren
Date Created	5th February 2022	Date Last Updated	5th February 2022

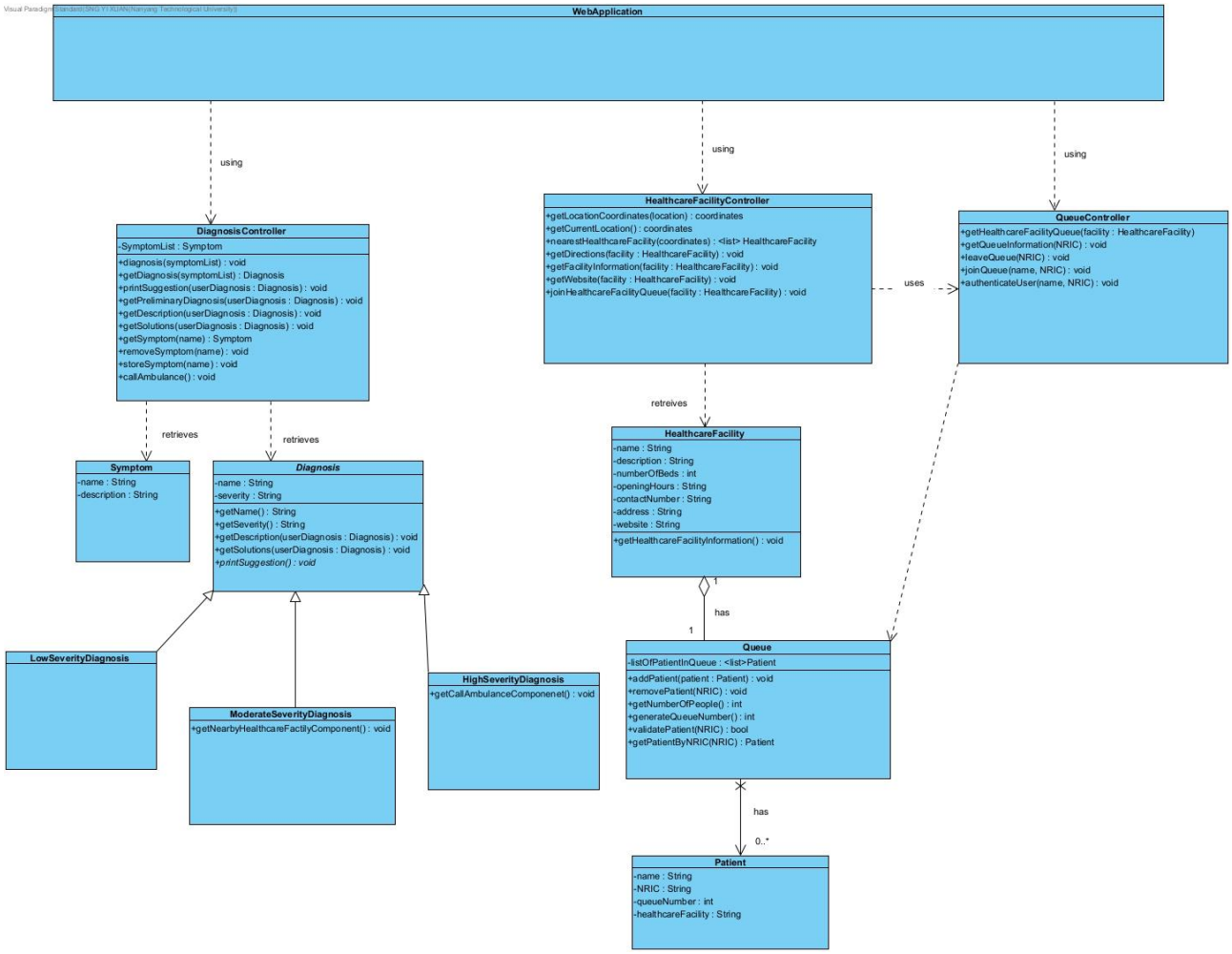
Actor:	User
Description:	As a user, I will like to be given the option to leave the queue
Precondition:	1. User is in the queue
Postcondition:	1. User's information is no longer in the healthcare facility's queuing database 2. Queue updated
Priority:	Moderate
Frequency of Use:	0-1 times per lifetime (Unlikely that that the user will want to leave a queue too many times, as the healthcare facilities being recommended are nearby already and probably the best choices available for the user)
Flow of Events:	1. User click Leave the queue in Queue information page 2. User confirm leaving the queue
Alternative Flow:	-
Exceptions:	EX1: If the user clicks leave queue when they have not joined any queue 1. The system displays the message "You have not joined any queue!" for 2 seconds.
Includes:	-
Special Requirements:	-
Assumptions:	1. Queue tracking can be achieved to monitor the real time queue situation with healthcare facility database
Note and Issues:	-

Use Case ID	12		
Use Case Name	Show possible routes to healthcare facility		
Created by	Ivan Lua	Last Updated by	Ivan Lua
Date Created	5th February 2022	Date Last Updated	5th February 2022

Actor:	User
Description:	When a user selects a healthcare facility, he is likely to want directions in the form of possible routes to his chosen healthcare facility
Precondition:	1. User has enabled GPS tracking or manually entered his address to get the nearest healthcare facilities 2. GoogleMaps API and Directions API must be running and connected to the website
Postcondition:	-
Priority:	Low
Frequency of Use:	1 time per lifecycle (User will only want to obtain a route to his selected healthcare facility)
Flow of Events:	1. User types address into search box or enable GPS and there are results of healthcare facilities within his radius of 5 kilometers 2. User hovers over a selected healthcare facility/clicks on the healthcare facility name on the selection pane 3. User clicks on “Get directions” button to receive possible routes from his current location to selected healthcare facility
Alternative Flow:	-
Exceptions:	-
Includes:	-
Special Requirements:	GPS hardware capability must be available in user’s device to enable automatic GPS tracking of his location

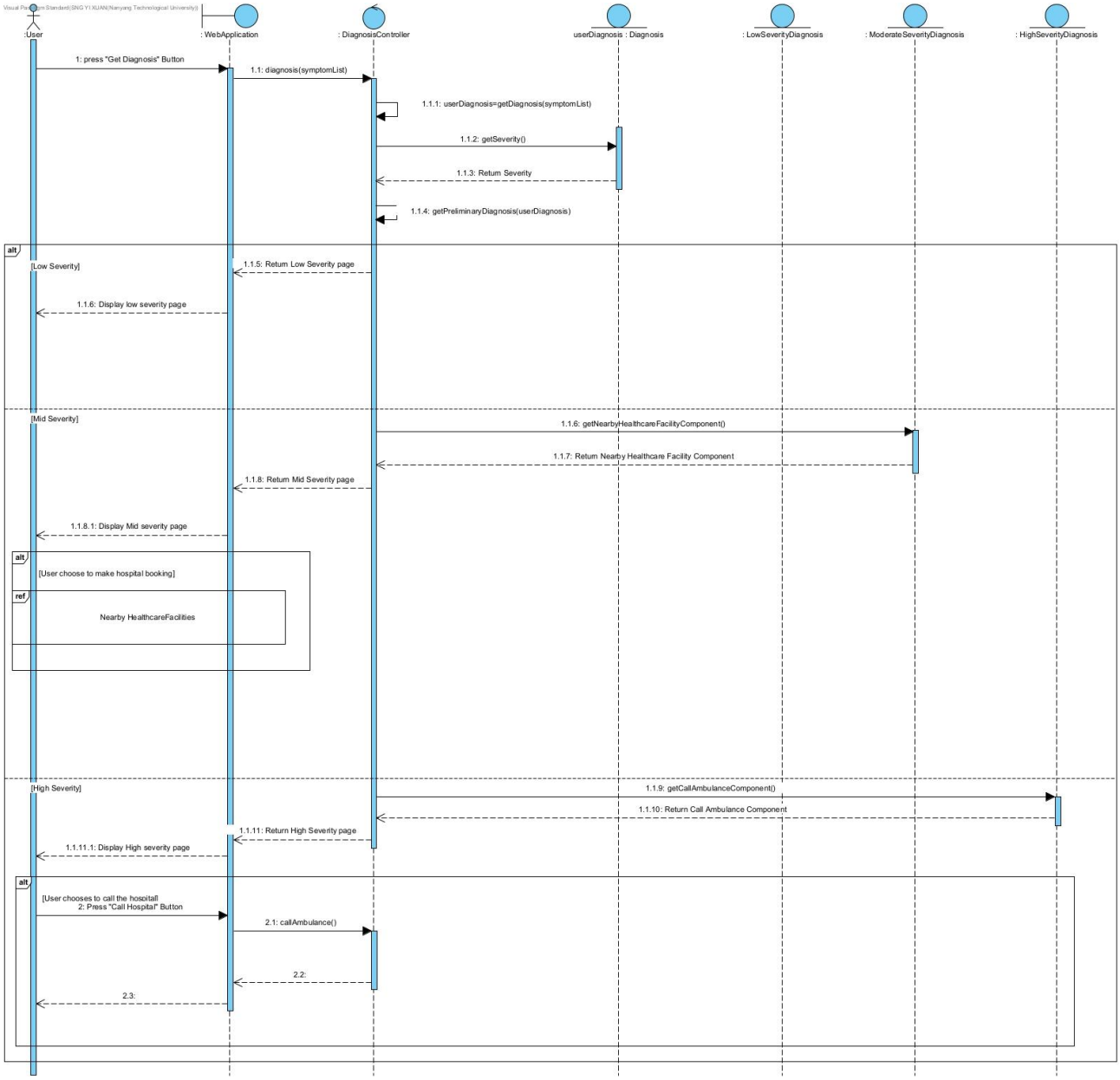
Assumptions:	There are healthcare facilities within a radius of 8 kilometers of the user's location
Note and Issues:	-

Class Diagram

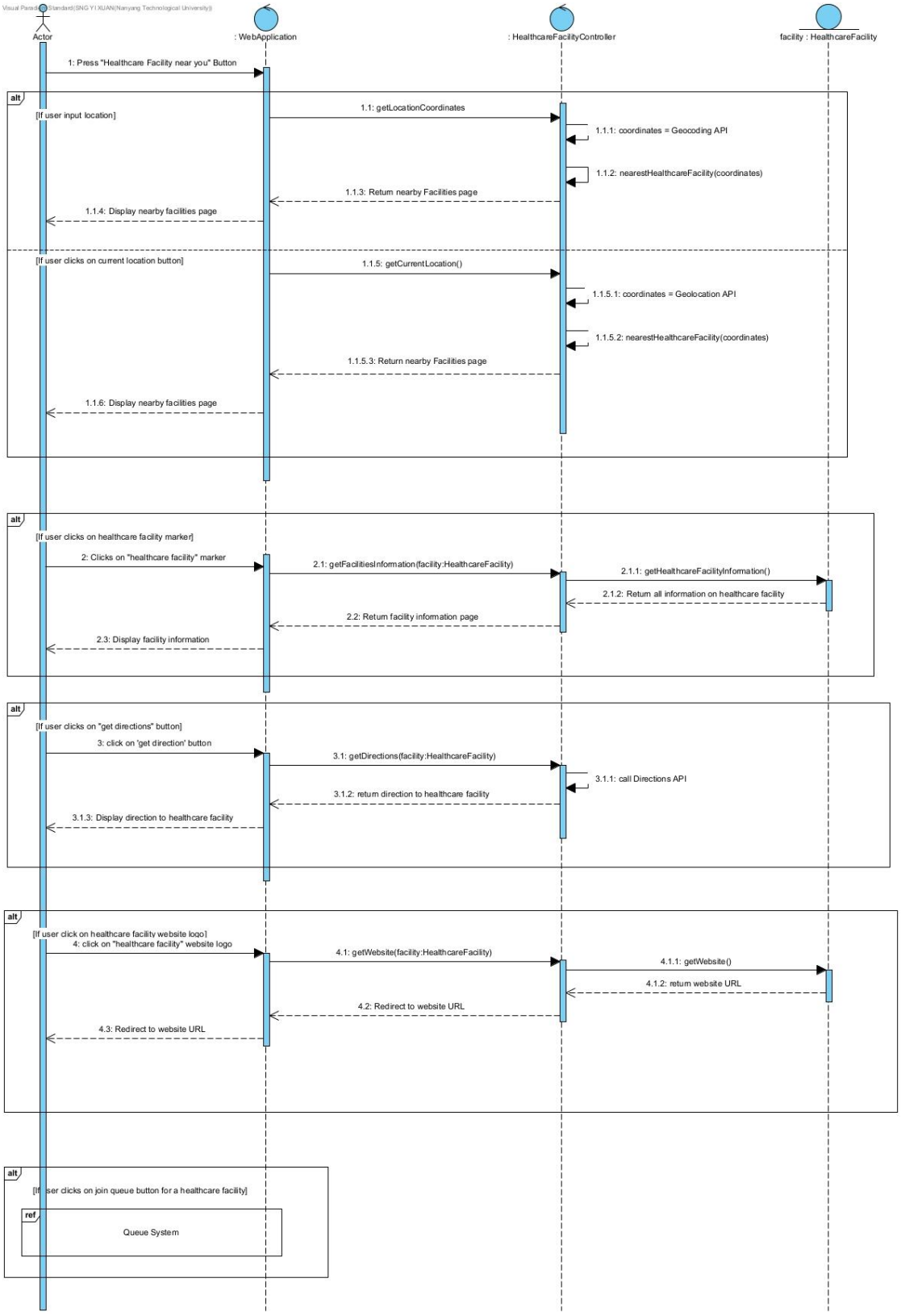


Sequence Diagram

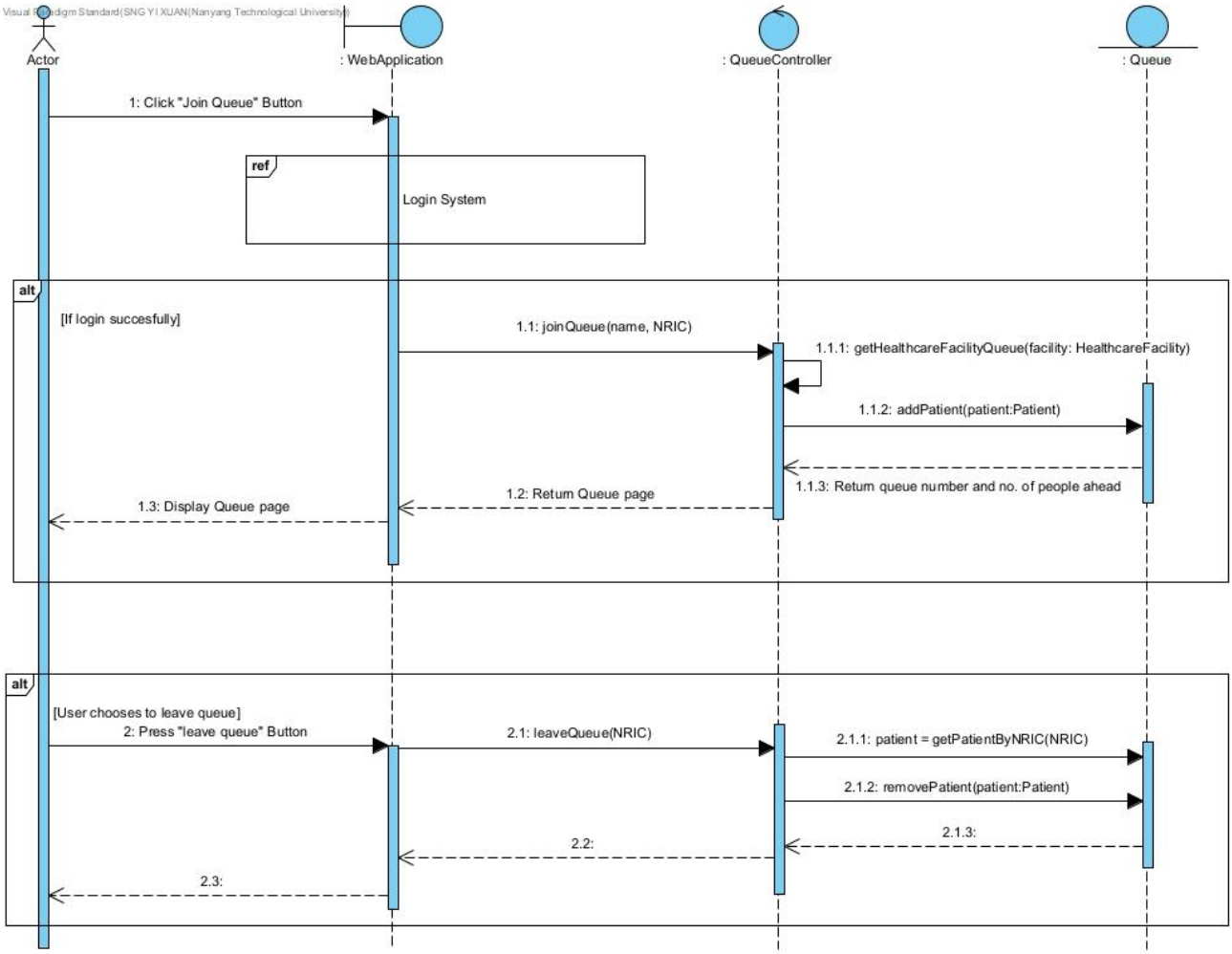
Diagnose



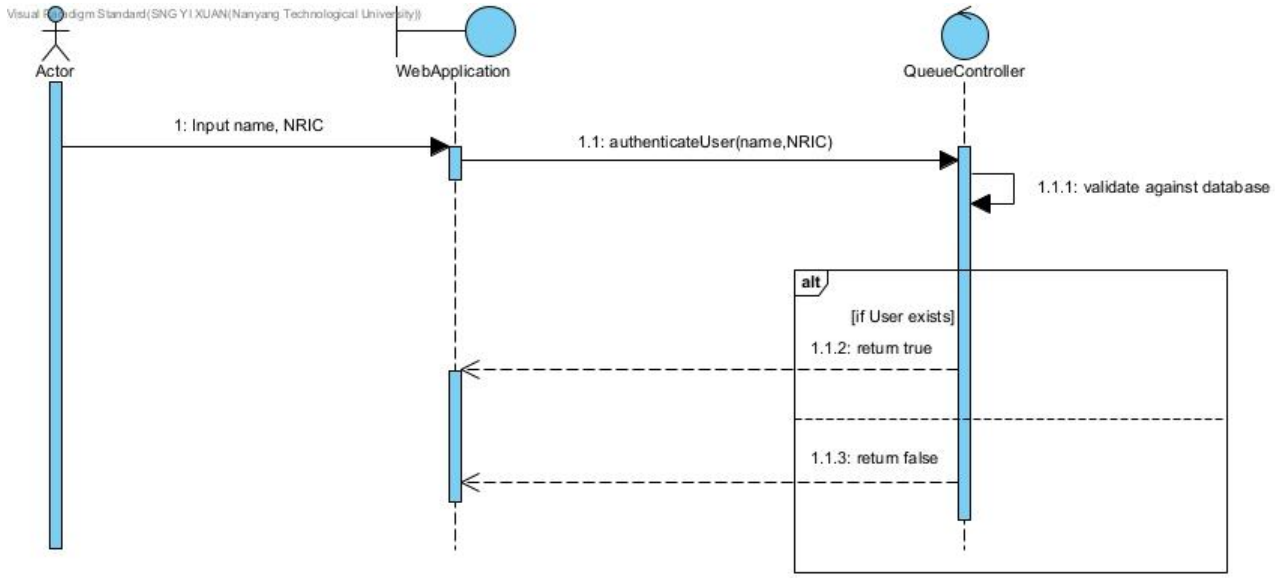
Nearby Healthcare Facilities



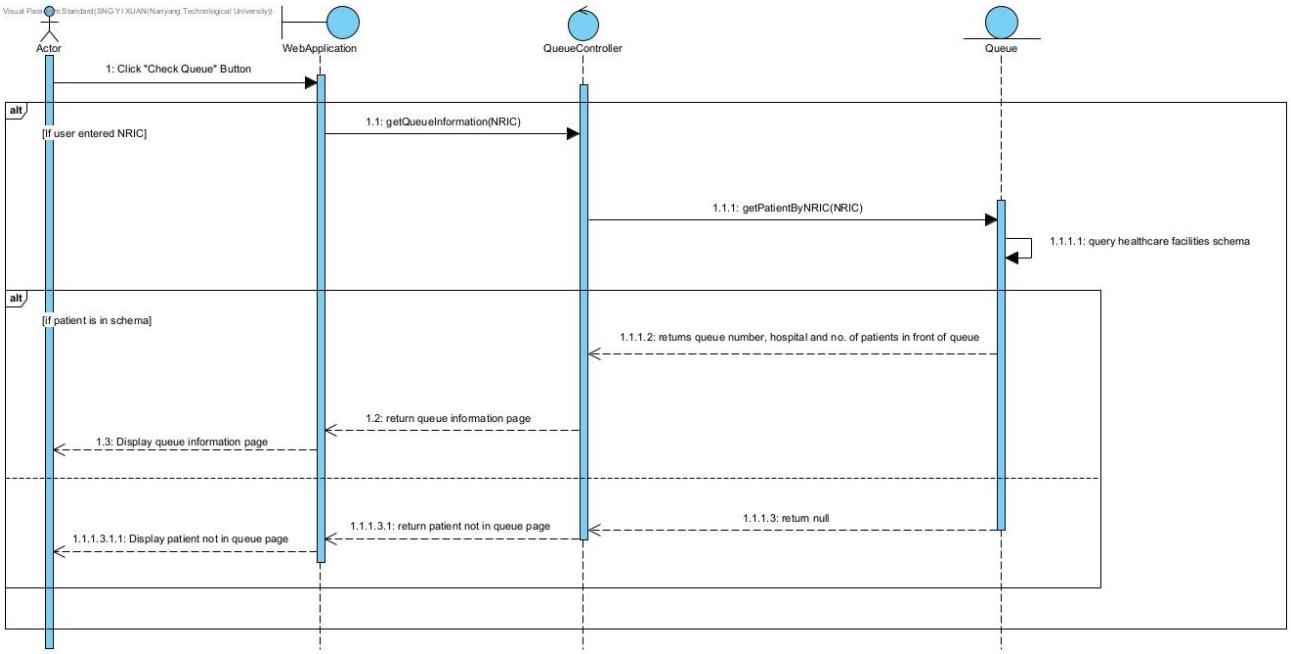
Queue System



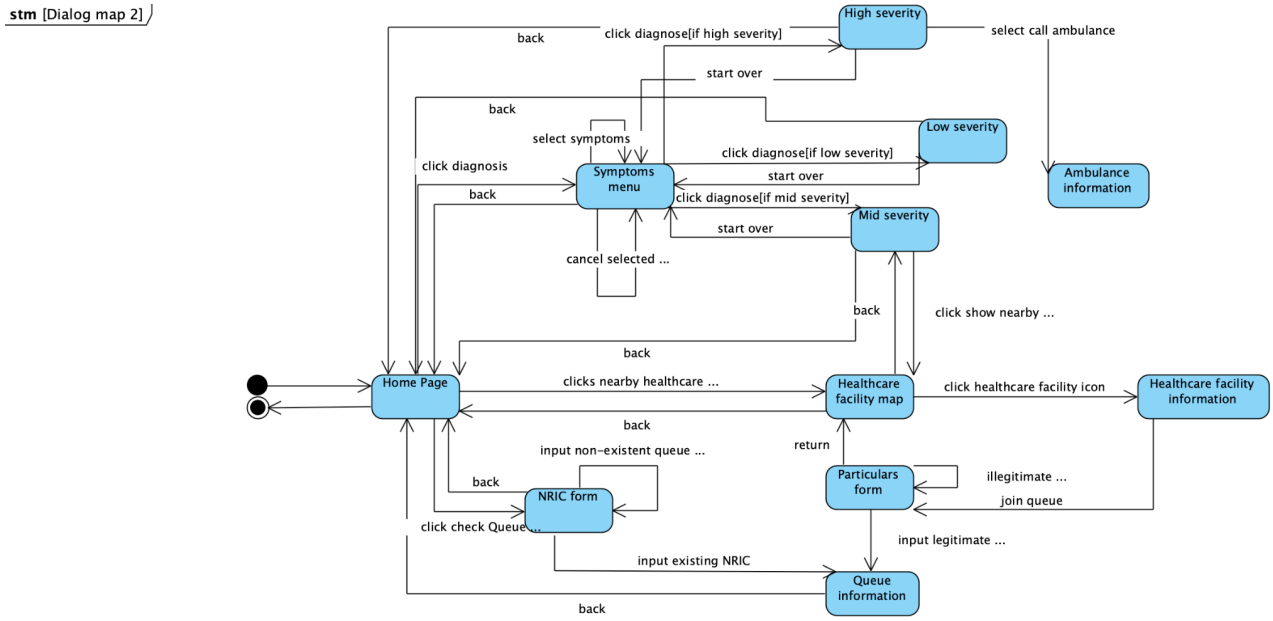
Login System



Check Queue



Dialog Map



Appendix C: To Be Determined List

Below is a list of future developments that our application aims to deliver:

Symptoms collection page:

- ➢ Show symptoms images
- ➢ Sidebar that allows the user to easily view, toggle and manage his searched symptoms

Queue management system:

- ➢ SMS implementation to notify users if there are only 5 patients ahead of the user
- ➢ Integration with Singpass for verification and access to user's personal data
- ➢ Access to each healthcare facility's queue management system allowing us to remotely access and manipulate it

General:

- ➢ AI chatbox to help user better navigate the application

Appendix D: Initial UI Mock Up

Link to figma prototype

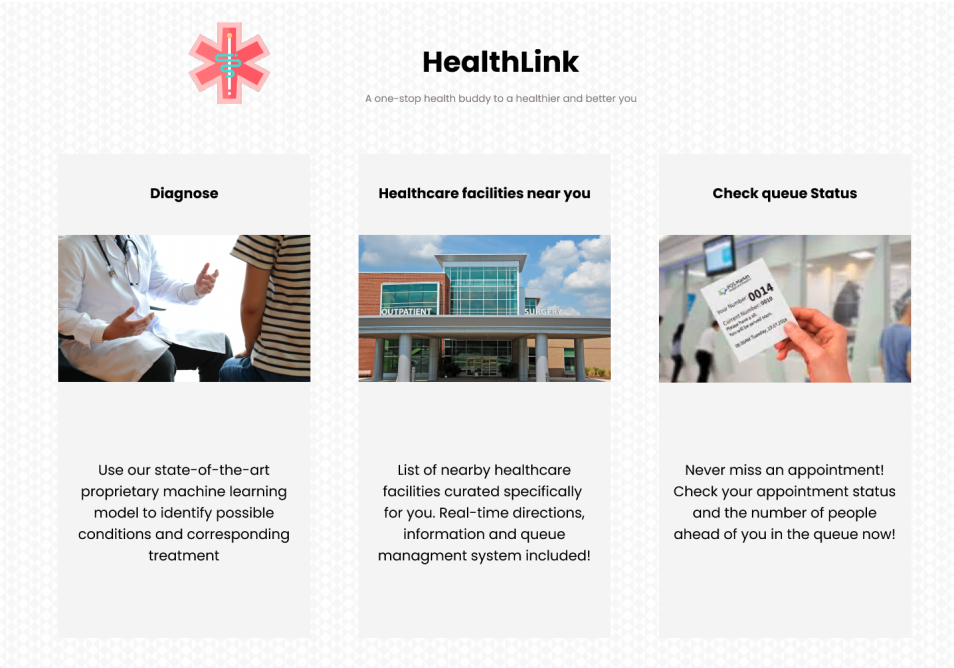


Figure 1: HealthLink Landing page

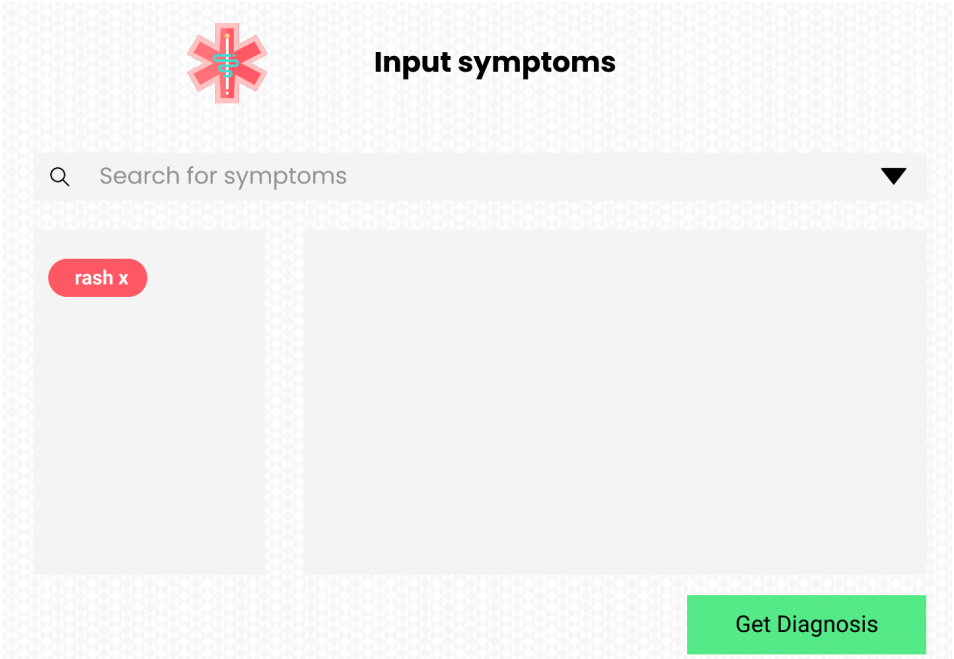


Figure 2: HealthLink symptoms input page

Figure 3: HealthLink low severity diagnosis page

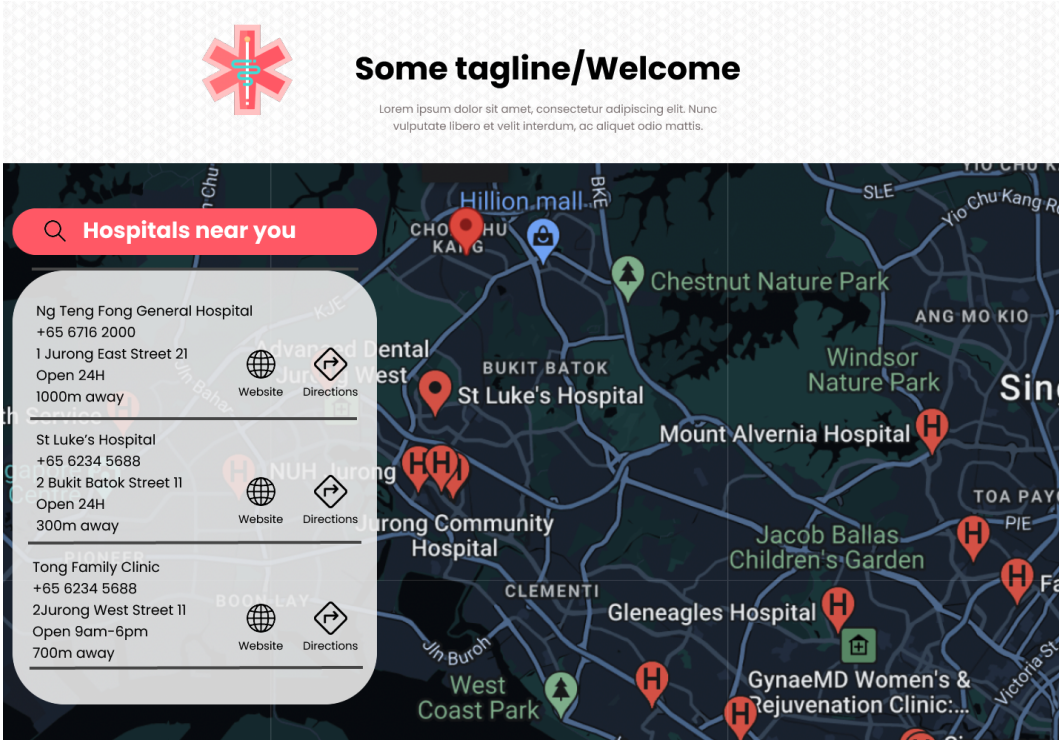


Figure 4: HealthLink healthcare facility near you page. From this page, user is able to find a healthcare facility to make a booking

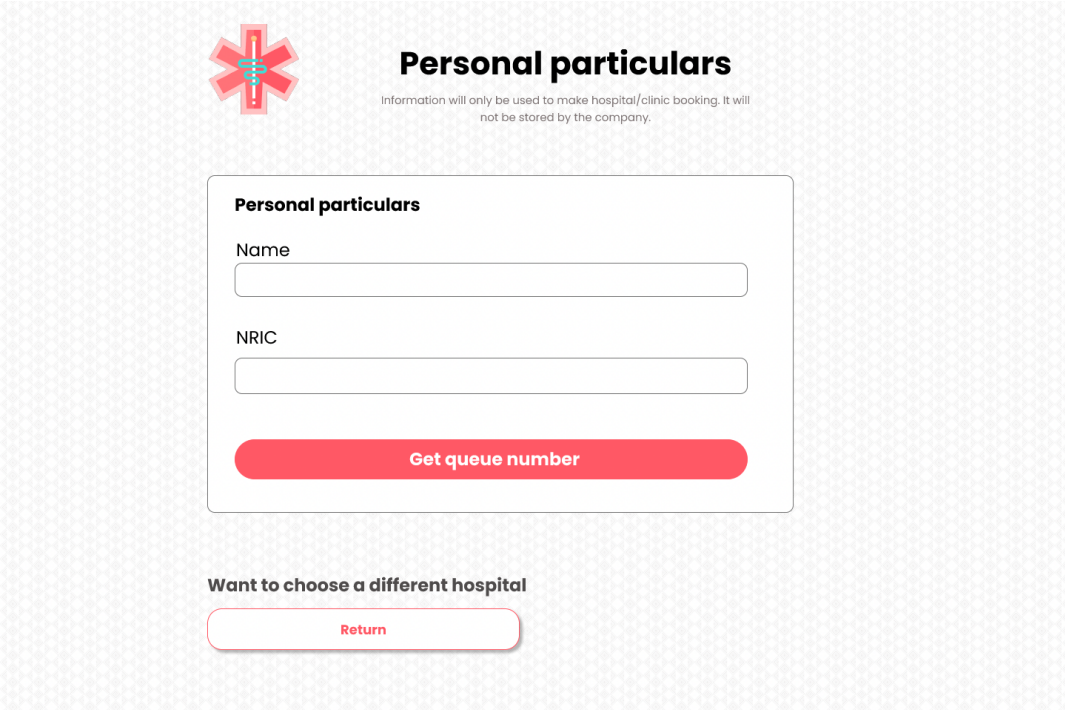


Figure 5: HealthLink input personal particular page. This is to create a healthcare facility booking for the user

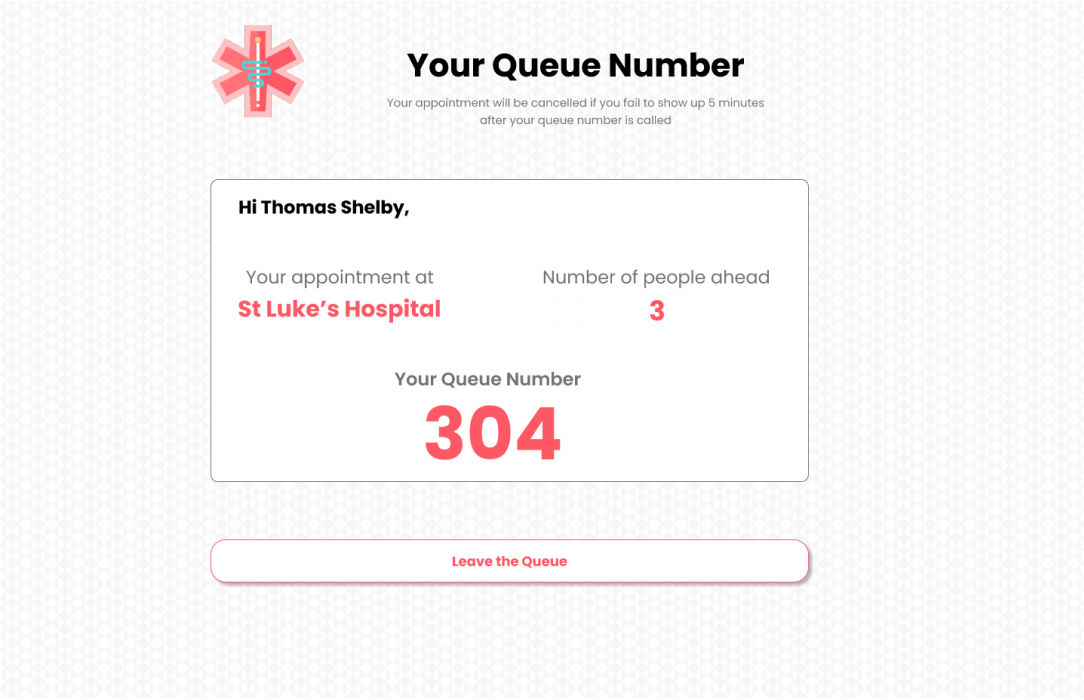


Figure 6: Queue number page. User is able to check the queue number of his booking

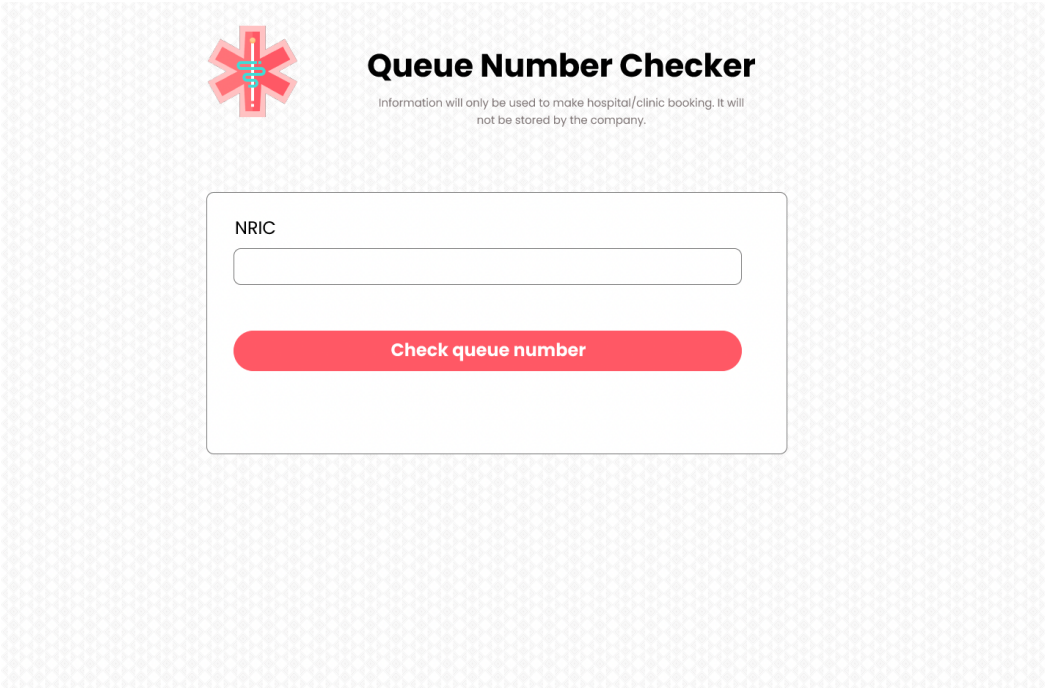
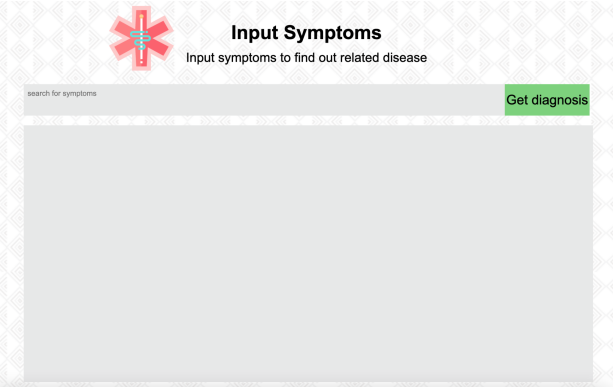
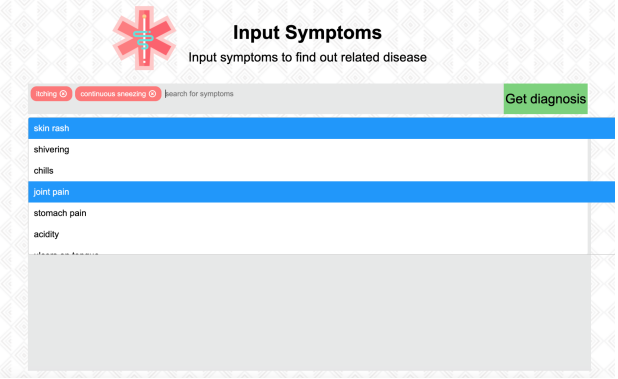
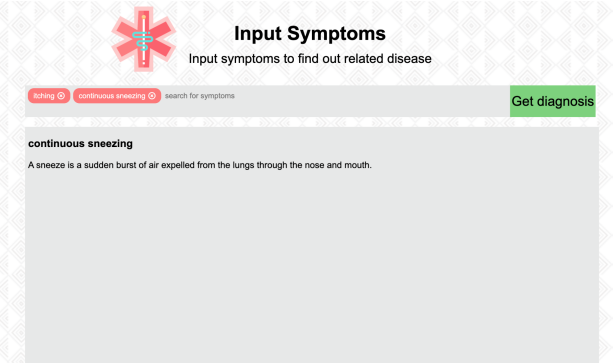
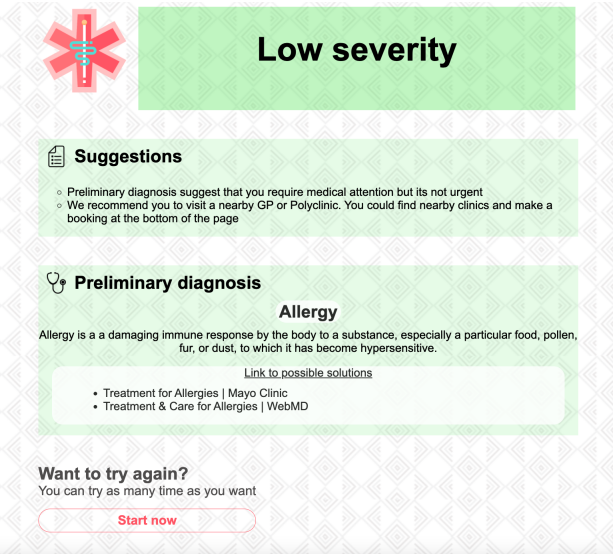
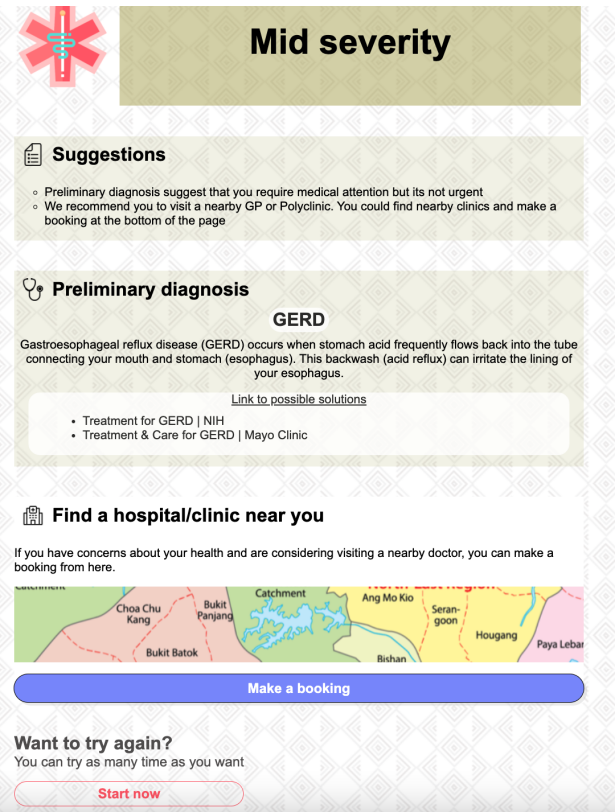


Figure 7: Queue number checker. From here, user is able to check their queue number information

Appendix E: User Manual

User View	User's Action
	Users open the HealthLink app to see the **main page**. Users can select one of three modules, **Diagnose**, **Hospitals near me**, and **Check queue status**, depending on whether they want to diagnose the symptoms they have, find hospitals near them, or check how many people are in front of them in a queue they've already joined respectively.
	User has selected the first option from the main page, **Diagnose**, and is now on the **input symptoms page**. The user can select the **search symptoms** box to view a drop down of available symptoms to pick from, or start typing a symptom by name.
	As the user enters their symptoms, all added symptoms will be displayed in the search for symptoms box. If the user has entered a symptom wrongly, they can select the cross button at the right of each symptom to remove it from the box.

	After the user enters each symptom, a description of said symptom will appear in the box below as well to help the user better evaluate if they actually have said symptom. Once the user has entered all known symptoms, they can select the **get diagnosis** box in green on the right, to get their diagnosis.
	Upon selecting the **get diagnosis** button, the page here on the left will be displayed. This is an example of the **low severity page**. Right on top, is the **severity** of the diagnosis our model has predicted based on the symptoms entered by the user. This will help determine the next course of action. Right below, Some **suggestions** will be provided on what are the actions the user can take next. Below that, a **preliminary diagnosis** box is provided, with the diagnosis obtained, a short summary of the diagnosis, and a few links to **self remedies** that the user can consider. Finally, there is a **start over** button, which the user can use if they want to try the diagnosis process again.



Upon selecting the get diagnosis button, the page here on the left will be displayed.

This is an example of the **mid severity** page.

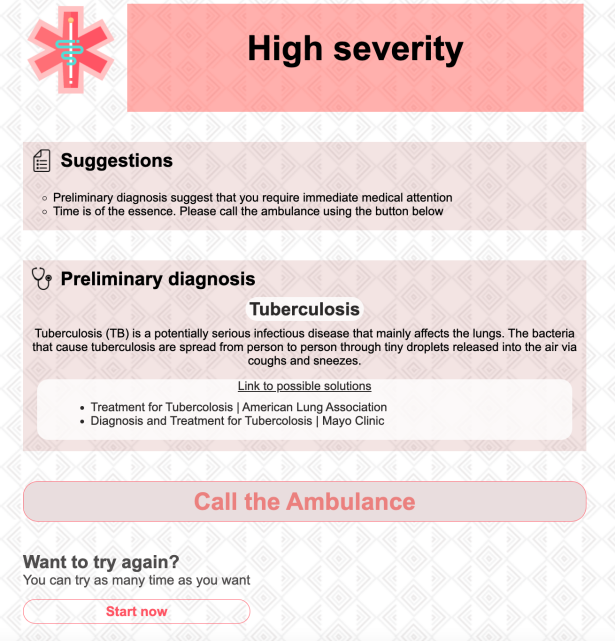
Right on top, is the **severity** of the diagnosis our model has predicted based on the symptoms entered by the user. This will help determine the next course of action.

Right below, Some **suggestions** will be provided on what are the actions the user can take next.

Below that, a **preliminary diagnosis** box is provided, with the diagnosis obtained, a short summary of the diagnosis, and a few links to **self remedies** that the user can consider.

For mid severity diagnosis, the app will next show a **find hospital/clinic** near you box where the user can select the make a booking button to proceed with finding a hospital and getting a queue number.

Finally, there is a **start over** button, which the user can use if they want to try the diagnosis process again.



Upon selecting the get diagnosis button, the page here on the left will be displayed.

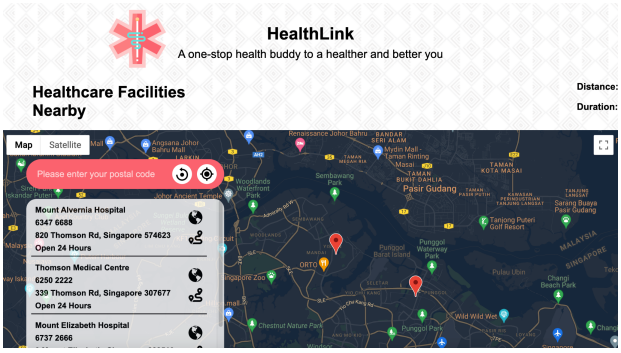
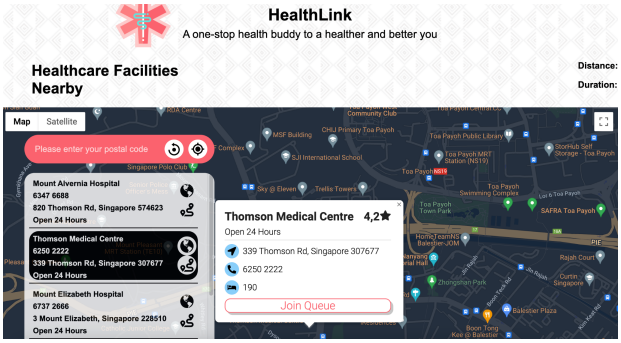
This is an example of the **high severity** page.

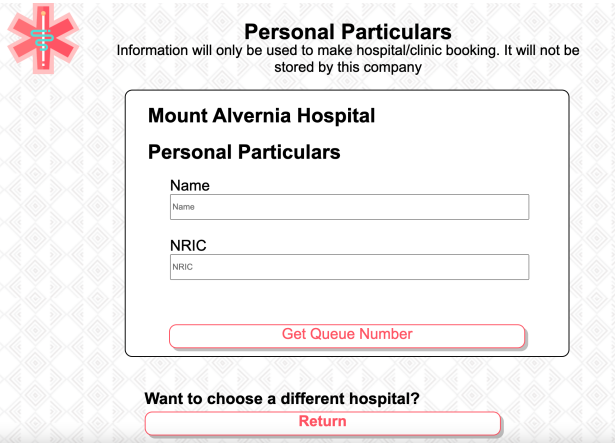
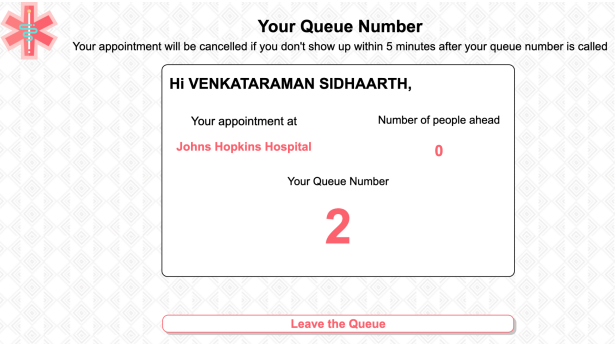
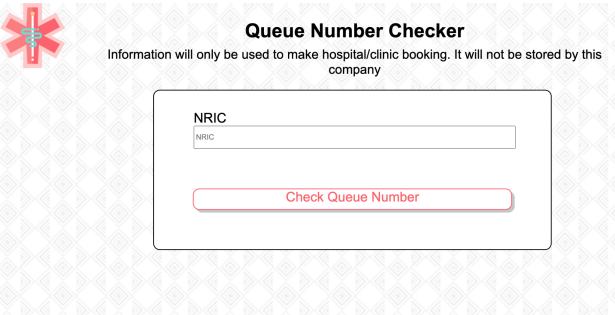
Right on top, is the **severity** of the diagnosis our model has predicted based on the symptoms entered by the user. This will help determine the next course of action.

Right below, Some **suggestions** will be provided on what are the actions the user can take next.

Below that, a **preliminary diagnosis** box is provided, with the diagnosis obtained, a short summary of the diagnosis, and a few links to self remedies that the user can consider.

For the high severity diagnosis, there will be a call to the ambulance button next, as the high

	severity means that the app has determined that the user likely needs **immediate medical attention**. Finally, there is a **start over** button, which the user can use if they want to try the diagnosis process again.
	User has selected the second option from the main page, **Healthcare facilities near you**. This page can be reached either by selecting the second module on the home page, or getting diagnosed with a **mid severity** diagnosis and clicking the make a booking button from the diagnosis page. The user can enter their postal code to find the hospitals nearest to that postal code. The user can also turn on their GPS to find the hospitals nearest to them. Alternatively, the user can scroll around on the map to manually select the hospital they want. The map can be zoomed in and out as needed.
	Upon clicking a particular hospital, the map will scroll to that particular hospital's location. The hospital will be highlighted in black in the menu on the left. There are 2 buttons displayed in the menu, the **globe button** links to the hospital website, and the button below shows the **shortest path** from the current location to the hospital that the user can take. A popup in white will be displayed from the **hospital's marker** on the map, with some relevant details about the hospital such as the contact details, address, opening hours, rating and the number of beds. At the bottom, there is a **join queue button which the user can select to join the hospital's queu**e.

	Upon selecting the join queue button, the user will be brought to the **personal particulars page**. The user will need to enter their **name and NRIC**. If the user enters an invalid NRIC, a popup showing 'invalid NRIC please try again' will be shown. After that, they will have to select **get queue number** button below the input fields to join the queue of the particular hospital. There is a **return button** available at the very bottom that the user can select if they decide they want to join a different hospital's queue as well. This button will take them to the nearest healthcare facility page again.
	Upon selecting the join queue button, the user will be brought to the **Queue number** page where their queue details will be shown. The details include their name, the hospital, the number of people ahead of them, and their queue number. This page can be accessed either by a user **entering their details and selecting join queue,** or later if the user wishes to check their details again and selects the third module from the main page, the **check queue module.** A **leave the queue** button is available at the bottom of the page if the user wishes to leave the queue at any point of time.
	User has selected the third option from the main page, **Check Queue Status,** and is now on the **Queue Number Checker page.** The user needs to enter their NRIC and select the **Check Queue Number** button to access the **personal particulars page** where the user can view their queue details as shown above. The user needs to have already joined a queue and have entered their correct NRIC, or a pop up saying invalid NRIC, check again! Will be shown.

Appendix F: Healthcare Facilities Near User Logic Flow Map