

CZ2006 Lab 1

SickGoWhere! App

GROUP A+

Members: Lee Xuanhui, Goh Shan Ying, Jonathan Chang,
Russell Leung, Lynn Masillamoni, Luke Chin Peng Hao

Contents

Functional requirements.....	2
Non-Functional requirements	5
Use Case Diagram.....	7
Use Case Descriptions.....	8
Data Dictionary	19
UI Mockup.....	20

Functional requirements

1. Nearest clinic/centre
 - 1.1. The system must be able to retrieve location of user's device via Geolocation API
 - 1.2. The system must be able to display the map with locations of the nearest clinics from the user
 - 1.3. The system must be able to retrieve locations of nearest CHAS clinics from data.gov.sg API
 - 1.4. The system must be able to recommend the nearest CHAS clinic based on distance from the user's current location
 - 1.5. The system must be able to redirect user to Google Maps Application to show direction to selected clinic
 - 1.6. The system must be able to store the user's favourite clinic/centre.
 - 1.7. The system must be able to display the clinic information
 - 1.7.1. The system must be able to display the clinic's location on a map
 - 1.7.2. The system must be able to display the clinic's location in a list
 - 1.7.3. The system must be able to display the operating hours of the clinic
 - 1.7.4. The system must be able to display the clinic's number and address
2. Taking queue number
 - 2.1. The system must be able to issue the user with a queue number
 - 2.1.1. The system must send an email confirmation to the user
 - 2.1.1.1. The email confirmation must contain the user's queue number for the selected clinic/centre
 - 2.1.1.2. The email confirmation must contain the estimated time that the user should reach the clinic/centre for their appointment
 - 2.1.2. The system must provide the user with the number of people that are ahead of them in the queue in real time
 - 2.1.3. The system must notify the user to make their way to the clinic when there are 3 people ahead of them in their queue
 - 2.1.3.1. The user shall be notified via an in app popup
 - 2.1.3.2. The user shall be notified via an email
 - 2.1.4. The user must be able to cancel their appointment
 - 2.1.4.1. The system must send an email confirmation to the user
 - 2.1.4.2. The system must remove the user from the clinic's waitlist

3. Login

- 3.1. The system must provide a login entry for the user and admin
 - 3.1.1. Users must be able to login using their email address and password
- 3.2. The system must be able to validate the user and admin's credentials with the database before allowing them into user/admin homepage
 - 3.2.1. The system must display an error message when the email is not registered
 - 3.2.1.1. The system shall prompt the user to create a new account.
 - 3.2.2. The system must display an error message when the password does not match with the user/admin password in the database
 - 3.2.3. Upon successful login, the system must redirect users to their homepage
 - 3.2.3.1. The admin must be redirected to the admin's personal homepage

4. Registration

- 4.1. The user must be able to register for a new account via the system registration
 - 4.1.1. System must prompt the user for the following information when registering for a new account:
 - 4.1.1.1. The user must enter his name
 - 4.1.1.2. The user must enter his email address
 - 4.1.1.3. The user must enter a strong password
 - 4.1.2. The system must be able to validate that all required fields have been filled up
 - 4.1.2.1. The system must be able to validate the account creation availability.
 - 4.1.2.2. The system must display the corresponding error message if validation fails.
 - 4.1.3. The system must redirect the user to the home page upon successful account creation

5. Admin

5.1. The admin must be able to view the list of registered users

5.1.1. The admin must be able to search for a specific user by searching for the email address

5.2. The admin must be able to delete a user account from the system

5.2.1. The System must display a dialog box asking 'Are you sure you want to remove this user?' with 2 options 'Confirm' and 'Cancel' to prevent accidental deletion of user account

5.2.1.1. If the admin selects 'Confirm', the system must delete the user's account from the system

5.2.1.1.1. The user must not be able to login after account deletion

5.2.1.2. If the admin selects 'Cancel', the system shall not do anything.

Non-Functional requirements

1. Usability Requirements

- 1.1. The system must reduce short term memory load
 - 1.1.1. The display must be simple and allow user to access the various functions easily
- 1.2. The system must permit easy reversal of actions
 - 1.2.1. The user must be able to return to the main menu or the previous page from the various functions of the app easily
- 1.3. The system must offer informative feedback
 - 1.3.1. The system shall provide necessary feedback to the user when an invalid input occurs
 - 1.3.2. The system shall provide appropriate error messages when a process fails
- 1.4. The system must strive for consistency
 - 1.4.1. A consistent sequence of actions is required for similar situations
 - 1.4.2. The display should not look drastically different across the different functions of the application

2. Performance Requirements

- 2.1. The system will not crash during the app's lifetime
- 2.2. The system must be able to run with little or no downtime
- 2.3. The system must have fast response times
- 2.4. The user must not experience any input lag or latency issues
- 2.5. The system must display the real time queue sequence for user's appointment
- 2.6. Email must be sent to users within 10 seconds

3. Compliance Requirements

- 3.1. The system must be able to run on ARM Android devices with application binary interface(ABI) of x86_64
- 3.2. The system must be able to support Android version of at least 7.0

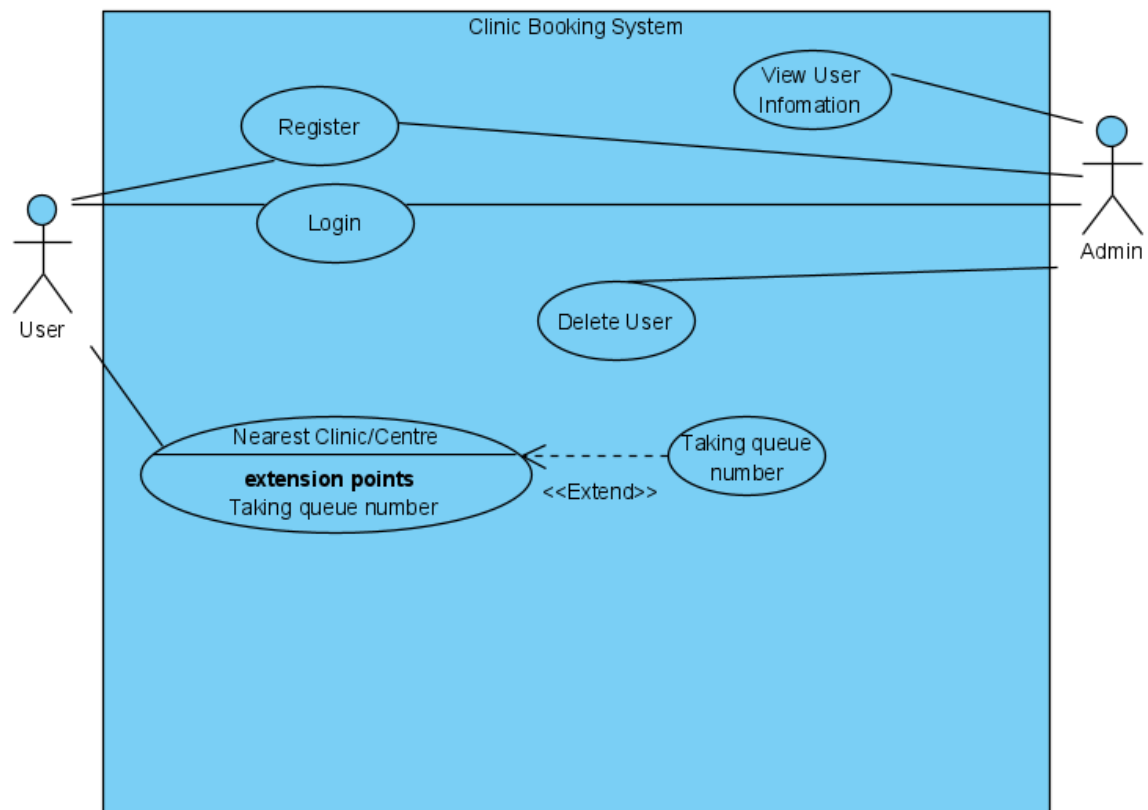
4. Security

- 4.1. The system must store Secure Hash Algorithm (SHA) hashed passwords into the database to deter hacking attempts
- 4.2. The system must mask the password field with asterisks during login to prevent snooping
- 4.3. The database must be confidential
 - 4.3.1. The user/ admin's information must be hidden from the public
 - 4.3.2. The data should not be used for other purposes other than storage under PDPA.

5. Reliability

- 5.1. When a tightly-coupled database is updated, all related database must be updated
 - 5.1.1. When user's database is updated, admin's database must be updated
 - 5.1.2. When admin's database is updated, user's database must be updated

Use Case Diagram



Use Case Descriptions

Primary Actors	Use cases
User	1.0 Register for a new account 5.0 Login 4.0 Nearest Clinic/Centre 6.0 Taking queue number
Admin	1.0 Register for a new account 5.0 Login 6.0 Taking queue number 2.0 View User Information 3.0 Delete User

Use Case ID:	1		
Use Case Name:	Register for a new account		
Created By:	Jonathan Chang	Last Updated By:	
Date Created:	2nd February 2021	Date Updated:	2nd February 2021

Actor:	User and Admin
Description:	Registration for a new user account
Preconditions:	<ol style="list-style-type: none"> 1. User/admin account must not already exist in user database 2. Mobile phone must be connected to the internet
Postconditions:	<ol style="list-style-type: none"> 1. A confirmation email will be sent to the user/admin's registered email address
Priority:	Medium

Frequency of Use:	1 - 3 times per lifetime
Flow of Events:	<ol style="list-style-type: none"> 1. User/admin enters valid email address, first name, last name, password and confirm password fields 2. User/admin selects register button 3. System validates that password and confirm password fields are identical 4. System validates current user account doesn't already exist in database 5. System sends confirmation email to the registered email address 6. System stores the new account information into the database
Alternative Flows:	<p>AF-S3: System detects mismatch between password and confirm password fields:</p> <ol style="list-style-type: none"> 1. System displays error message "Password does not match field above, please enter password again" 2. Return to step 1 <p>AF-S4: System detects that user/admin account already exists in database:</p> <ol style="list-style-type: none"> 1. System displays an error message "This user already exists!" 2. Return to step 1
Exceptions:	
Includes:	
Special Requirements:	
Assumptions:	
Notes and Issues:	

Use Case ID:	2		
Use Case Name:	View User Information		
Created By:	Jonathan Chang	Last Updated By:	
Date Created:	2nd February 2021	Date Updated:	2nd February 2021

Actor:	Admin
Description:	Ability to view user information
Preconditions:	Must login with admin credentials
Postconditions:	-
Priority:	Medium
Frequency of Use:	1-5 times a year
Flow of Events:	<ol style="list-style-type: none"> Admin views the email addresses and names of all the users in the database by scrolling down the list
Alternative Flows:	AF1- Admin searches for a specific user's email address <ol style="list-style-type: none"> The email address of each user that corresponds to the search will be shown
Exceptions:	
Includes:	
Special Requirements:	
Assumptions:	
Notes and Issues:	

Use Case ID:	3		
Use Case Name:	Delete User		
Created By:	Jonathan Chang	Last Updated By:	
Date Created:	2nd February 2021	Date Updated:	2nd February 2021

Actor:	Admin
Description:	Ability to delete user account from database
Preconditions:	<ol style="list-style-type: none"> 1. Must login with admin credentials 2. User to be deleted must exist in the database
Postconditions:	Users that have had their account removed from the database cannot log in using the previously registered email address anymore
Priority:	Medium
Frequency of Use:	1-5 times a year
Flow of Events:	<ol style="list-style-type: none"> 1. Admin clicks on the red cross next to the user account that they wish to remove 2. System displays a dialog box asking 'Are you sure you want to remove this user?' with 2 options 'Confirm' and 'Cancel' 3. Admin selects 'Confirm' 4. System sends an email to the user's registered email informing them that their account has been removed 5. System removes user data from the user database
Alternative Flows:	AF-S2: If admin selects 'Cancel', the user

	account selected will not be removed and will remain in the database
Exceptions:	
Includes:	
Special Requirements:	
Assumptions:	
Notes and Issues:	-

Use Case ID:	4		
Use Case Name:	Nearest Clinic/Centre		
Created By:	Lee Xuanhui	Last Updated By:	
Date Created:	2nd February 2021	Date Updated:	2nd February 2021

Actor:	User
Description:	Display favourite clinics/centres and find nearest 5 clinics/centres to the user
Preconditions:	<ol style="list-style-type: none"> 1. User is signed in 2. Mobile phone must have GPS enabled 3. Mobile phone must be connected to the internet
Postconditions:	<ol style="list-style-type: none"> 1. Favourite clinics/centres will be displayed 2. 5 Nearest clinics/centres will be displayed
Priority:	High
Frequency of Use:	

Flow of Events:	<ol style="list-style-type: none"> 1. User clicks on the “Nearby Clinics/Centres” button 2. System displays user’s favorites clinics/centres on the map 3. System displays top 5 nearest clinics/centres on the map
Alternative Flows:	<p>AF1: If the user clicks on the “List view” button</p> <ol style="list-style-type: none"> 1. System will display both user favourites and top 5 nearest clinics/centres in a top-down list <p>AF2: If the user does not have any favourite clinics/centres</p> <ol style="list-style-type: none"> 1. System will just display the top 5 nearest clinics/centres only
Exceptions:	<p>EX1: GPS is not turned on</p> <ol style="list-style-type: none"> 1. Pop up message appears for user to enable GPS
Includes:	
Special Requirements:	
Assumptions:	
Notes and Issues:	

Use Case ID:	5		
Use Case Name:	Login		
Created By:	Goh Shan Ying	Last Updated By:	
Date Created:	2nd February 2021	Date Updated:	2nd February 2021

Actor:	User and Admin
Description:	Login to SickGoWhere! App
Preconditions:	<ol style="list-style-type: none"> 1. User/admin must have the app downloaded on their phone 2. User/admin must be connected to the internet
Postconditions:	<ol style="list-style-type: none"> 1. User/admin is logged in 2. User/admin is in its Home page
Priority:	High
Frequency of Use:	Everytime the app is used
Flow of Events:	<ol style="list-style-type: none"> 1. System will display Login page 2. User/admin enters a valid username and password field 3. User/admin clicks the login button 4. System validates current User/admin account exists in database 5. System validates password match with the current User/admin account's password in the database 6. Upon successful login, the system will redirect user to the user's/admin's homepage
Alternative Flows:	AF1-S1: User choose to register for a new account by clicking the "Register" button the Login page:

	<ol style="list-style-type: none"> 1. Redirects user to registration page <p>AF2-S3: System detects empty fields:</p> <ol style="list-style-type: none"> 1. System displays error message "Please fill up both username and password field" 2. Return to step 2 <p>AF3-S4: System detects invalid username/ user account does not exists in database:</p> <ol style="list-style-type: none"> 1. System displays an error message "Account does not exist. Please register for a new account!" 2. System redirect back to Login page 3. AF1-S1 <p>AF4-S5: System detects that password provided by user does not match with the current User/admin account's password in the database:</p> <ol style="list-style-type: none"> 1. System displays an error message "Invalid username/ password!" 2. Retry getting password up to 3 times 3. If user still fails to provide correct password, report failure 4. Wait one minute before next login
Exceptions:	
Includes:	
Special Requirements:	
Assumptions:	
Notes and Issues:	

Use Case ID:	6
Use Case Name:	Taking queue number

Created By:	Lynn Masillamoni	Last Updated By:	
Date Created:	2nd February 2021	Date Updated:	2nd February 2021

Actor:	User
Description:	The user makes an appointment at a clinic/center of their choice. If successful, the system informs the user of their queue number and the estimated time the user has to reach the clinic/centre. The system displays in real time the number of people ahead of the user in the queue, and will also prompt the user to make their way to the clinic when there are 3 people ahead of them in the queue.
Preconditions:	<ol style="list-style-type: none"> 1. The user requests for an appointment at the clinic/centre of their choice
Postconditions:	<ol style="list-style-type: none"> 1. The system displays the number of people ahead of the user in the queue in real time 2. The system notifies the user to make their way to the clinic/centre when there are 3 people ahead of them in the queue through an email and a pop-up in the app
Priority:	High
Frequency of Use:	Approximately every time the app is used
Flow of Events:	<ol style="list-style-type: none"> 1. The user requests for an appointment at a clinic/centre of their choice 2. The system verifies that this request occurred during the clinic/centre's opening hours 3. The system adds the user into the clinic/centre's patients waitlist 4. The system generates the user's queue number

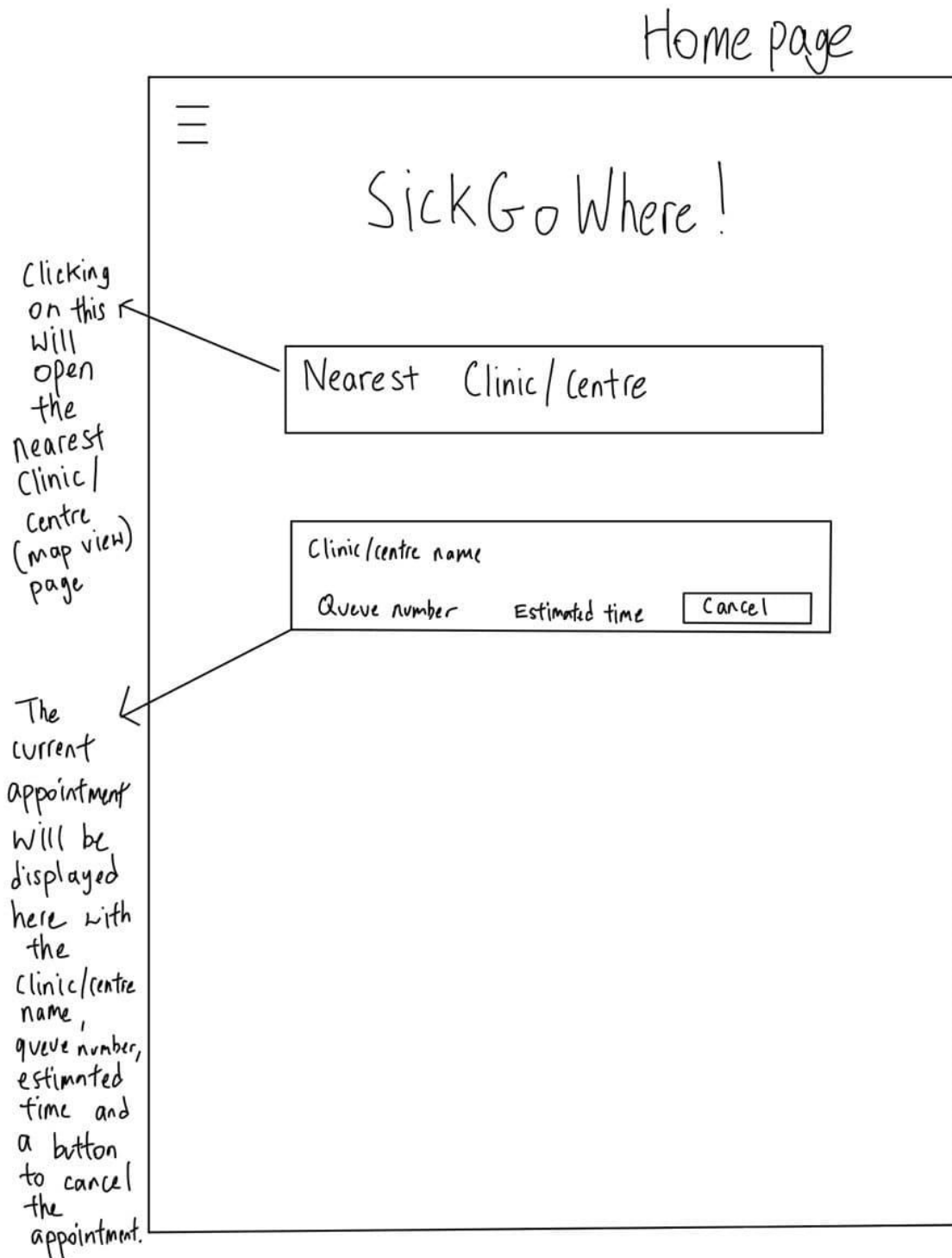
	<ol style="list-style-type: none"> 5. The system generates an estimated time that the user should reach the clinic/centre for the appointment 6. The system verifies that the estimated time that the user should reach the clinic/centre for the appointment is during the clinic/centre's opening hours 7. The system sends an email confirmation to the user with the user's queue number and estimated time the user should reach the clinic/centre 8. The system displays the number of people ahead of the user in the queue in real time in the app 9. The system notifies the user to make their way to the clinic/centre when there are 3 people ahead of them in the queue through an email and a pop-up in the app
Alternative Flows:	<p>AF-S2: Request for appointment occurs during the clinic/centre's closing hours</p> <ol style="list-style-type: none"> 1. The user receives an email indicating that the appointment was unsuccessful <p>AF-S6: The estimated time that the user should reach the clinic/centre occurs during the clinic/centre's closing hours</p> <ol style="list-style-type: none"> 1. The user is removed from the clinic/centre's patients waitlist 2. The user receives an email indicating that the appointment was unsuccessful
Exceptions:	<p>EX1: The user cancels the appointment before the estimated time that the user should reach the clinic/centre for the appointment</p> <ol style="list-style-type: none"> 1. The user is removed from the clinic/centre's patients waitlist 2. The user receives an email indicating that the appointment was successfully cancelled

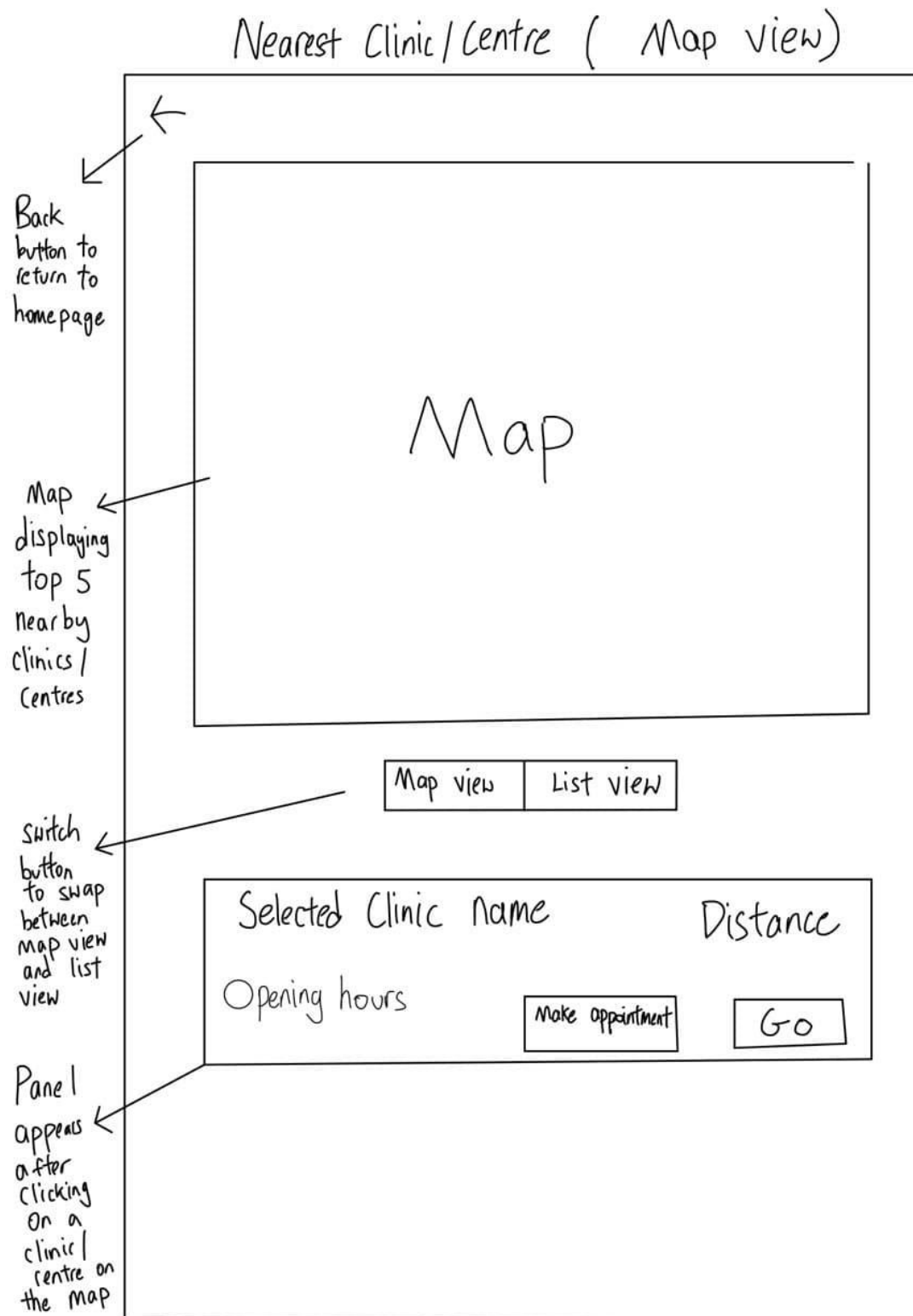
Includes:	
Special Requirements:	
Assumptions:	
Notes and Issues:	

Data Dictionary

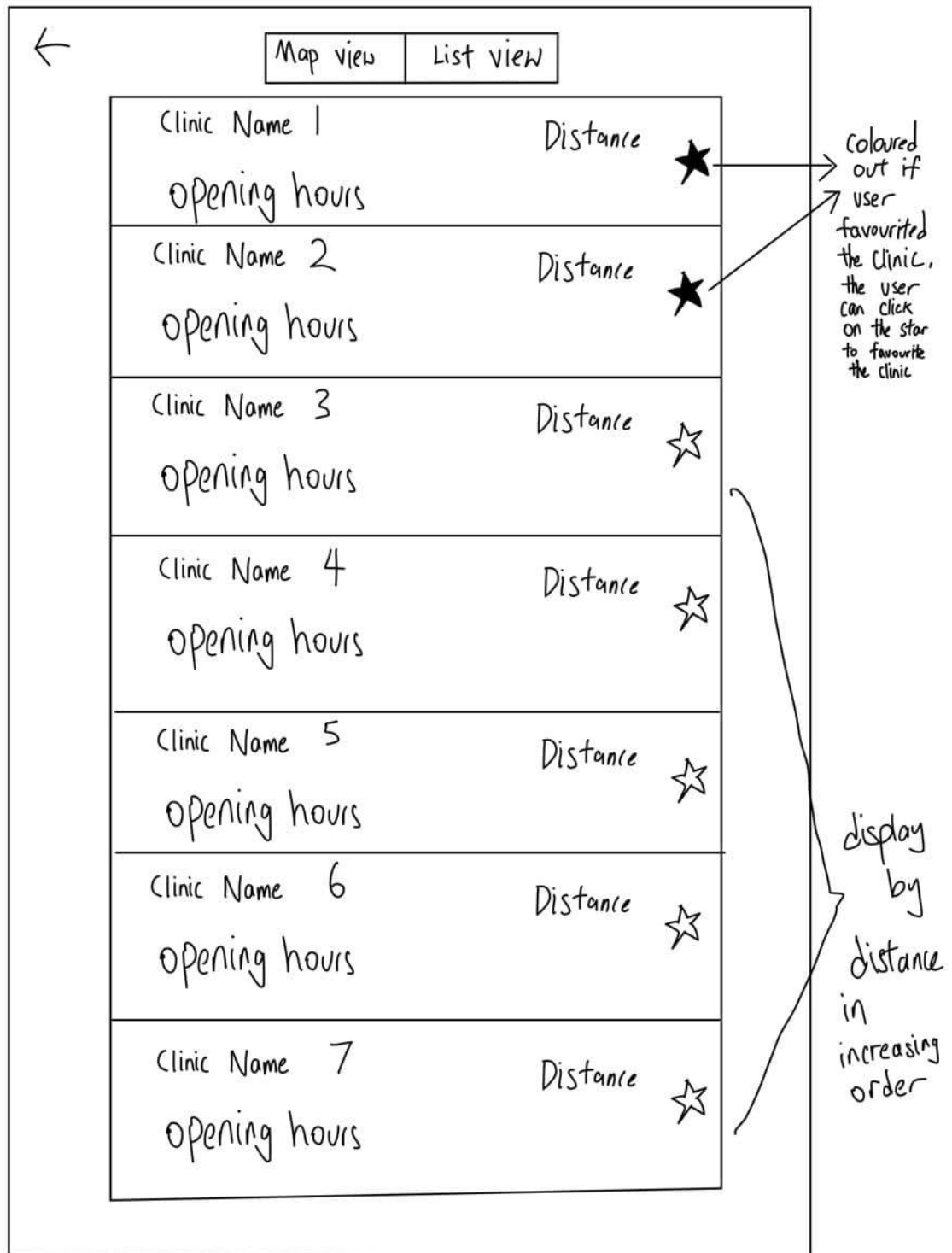
Term	Definition
Pharmacy	A shop or hospital dispensary where medicinal drugs are prepared or sold
Symptoms	A physical or mental feature which is regarded as indicating a condition of disease, particularly such a feature that is apparent to the patient
Clinic/centre	A place where patients are given medical treatment or advice
Queue number	Shows the number of people including the user in the queue
Appointment	An arrangement to meet a doctor at a clinic/centre with an assigned queue number
Patient	A person receiving or registered to receive medical treatment
Medical Consultation/appointment	A two-way social interaction where the doctor elicits information from the patient, then offers a diagnosis or opinion and may also discuss and offer treatment.
Doctor	A person who is qualified to treat people who are ill
User	A User is the default name for users who are not an Admin user. The user may have limited access to viewing and changing certain items in the SickGoWhere! app.
Admin	Admin is a user with additional permissions. Admins can delete and view user's information
GPS	Global positioning system- navigational service to determine user's location

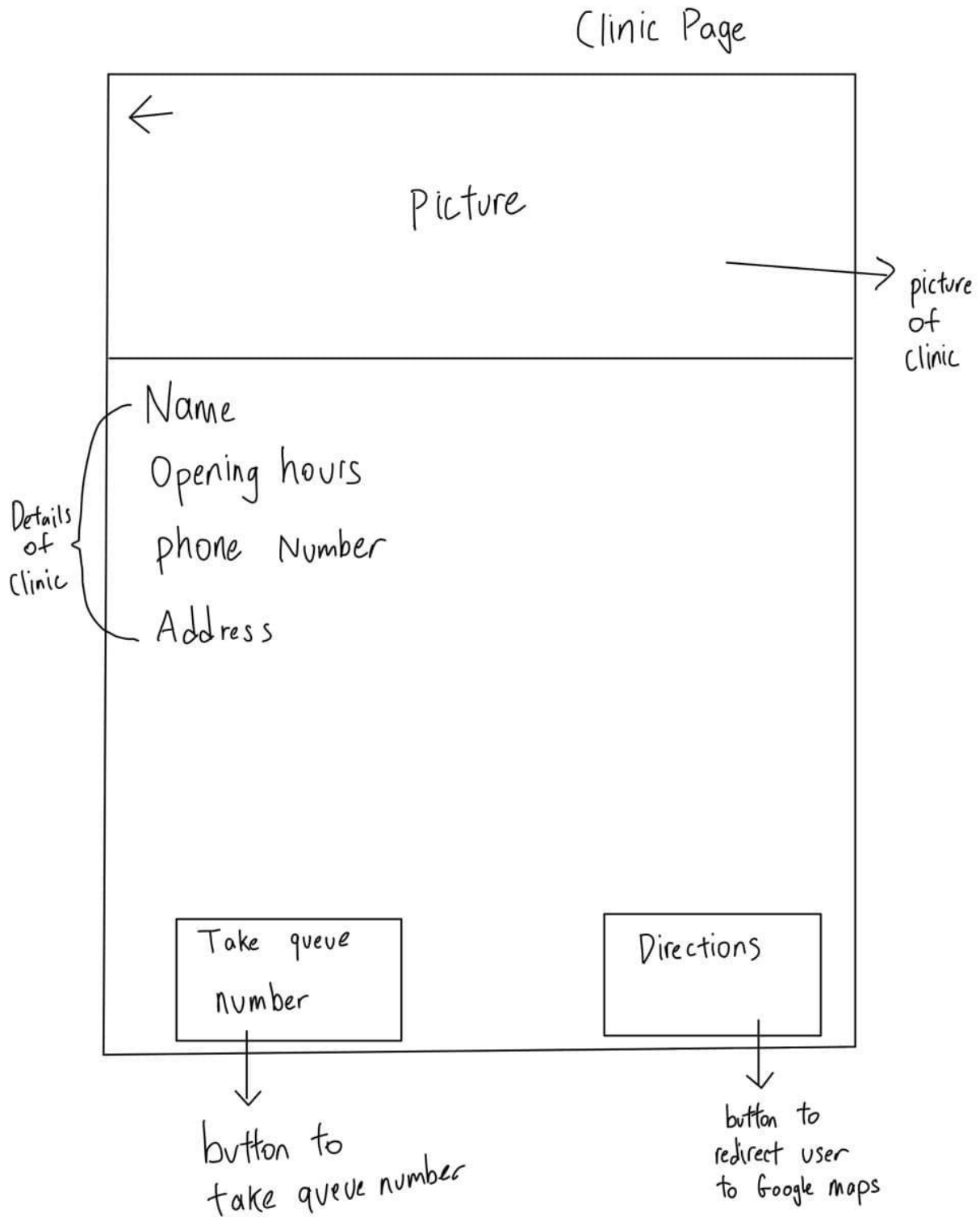
UI Mockup



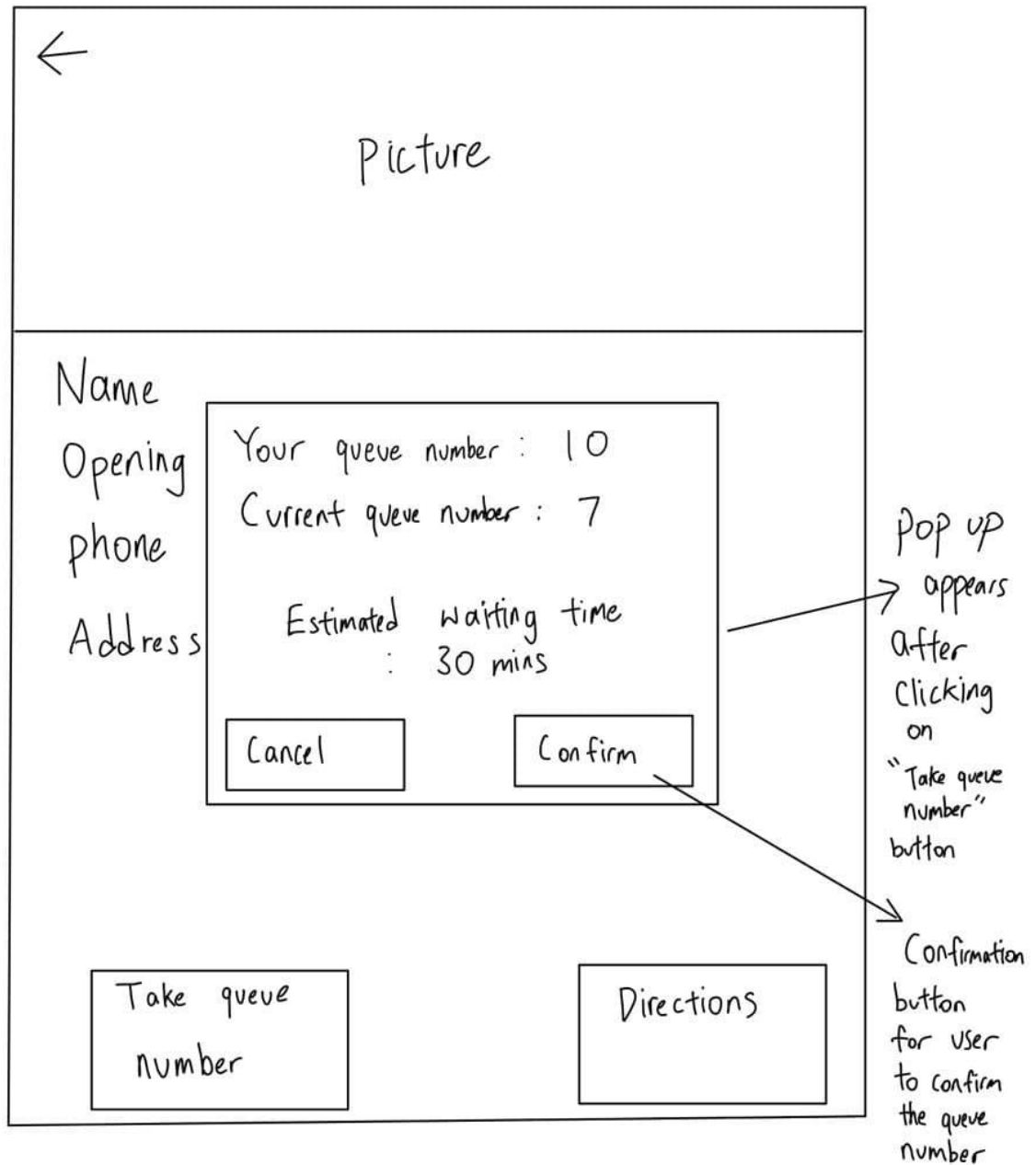


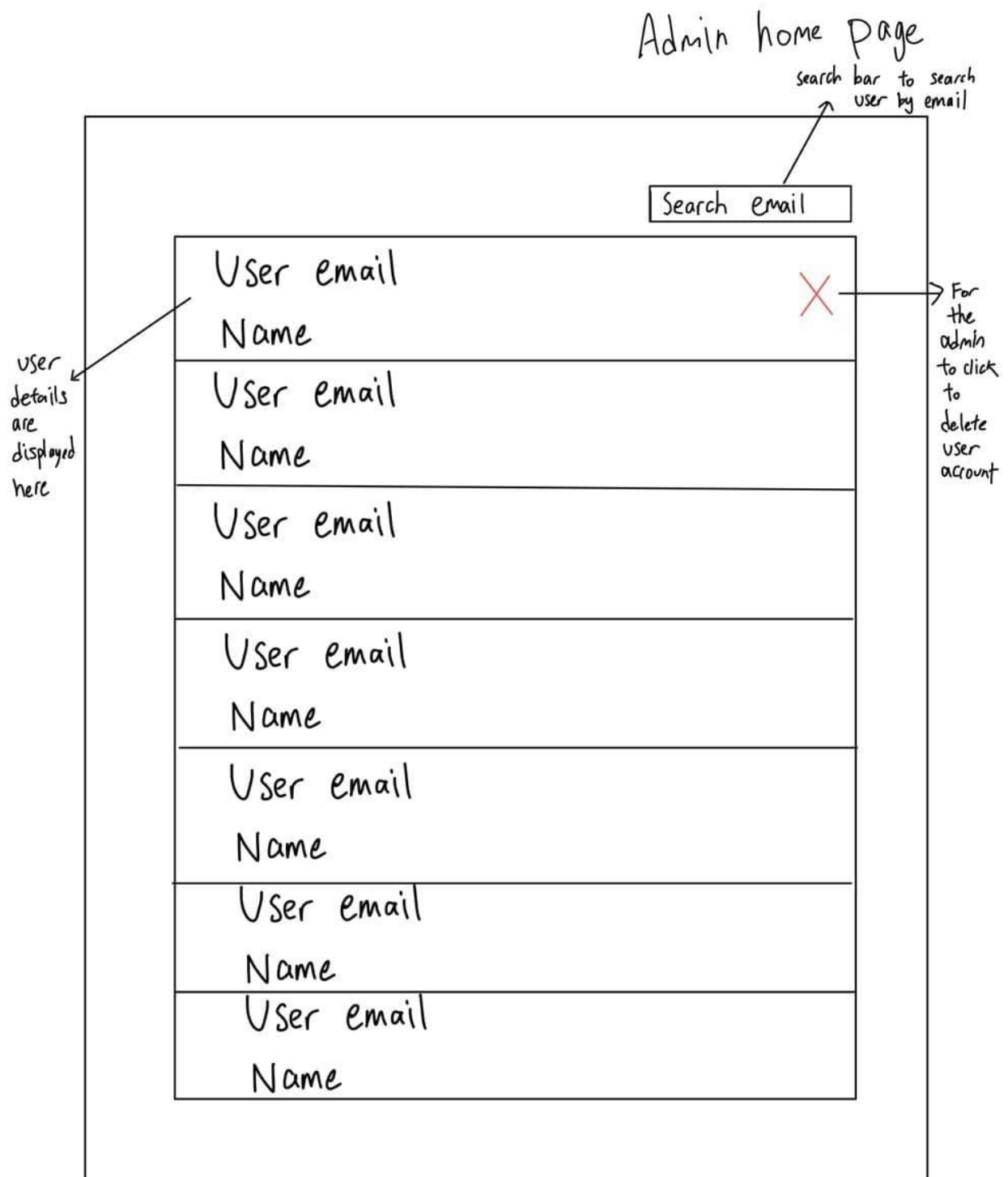
List view





Clinic Page with pop-up





Admin home page with pop up

Search email	
User email	X
Name	
User email	
Name	
User email	
Name	
User e	<div>Are you sure you want to remove this user ?</div> <div><input type="button" value="Cancel"/> <input type="button" value="confirm"/></div>
Name	
User email	
Name	
User email	
Name	
User email	
Name	

pop up to confirm user account deletion