

Problem Statement

Tinnitus, by definition, according to Mayo Clinic, is the perception of noise or ringing in the ears. Tinnitus is a common symptom that occurs to 15 to 20 percent of people and some of the symptoms include ringing, buzzing, clicking, hissing, humming, or roaring in the ears. As a result, some people may experience sleep problems, memory problems, concentration issues, depression, or even anxiety. Although these symptoms negatively affect quality of life, tinnitus is not an indication of a serious disorder and it can be improved through treatment. There are two kinds of tinnitus: subjective and objective tinnitus. Subjective tinnitus is the most common tinnitus and it is the tinnitus that can only be heard by oneself. Contrarily, objective tinnitus is a tinnitus that can be heard by the doctor when the doctor does an examination. There are many reasons that may cause tinnitus, and some of the most common reasons include hearing loss that comes from aging, long-term exposure to loud noise, and earwax blockage. Other less common reasons include meniere's disease, acoustic neuroma, eustachian tube dysfunction, muscle spasms in the inner ear, head injuries, and neck injuries. It also has been known that blood vessel disorders are linked to tinnitus. Some of the blood vessel disorders that cause tinnitus include atherosclerosis, high blood pressure, turbulent blood flow, and malformation of capillaries. Additionally, some medications that may cause tinnitus include antibiotics such as polymyxin B, erythromycin, and neomycin; cancer medications including cisplatin and methotrexate; water pills such as bumetanide and ethacrynic acid; and extremely high doses of aspirin (Sandhya, 2019).

Tinnitus Retraining Therapy (TRT) is a retraining method developed in the USA and England that prioritizes making the tinnitus disappear from perception by habituation. This means that through tinnitus retraining therapy, patients who are usually bothered by tinnitus will no longer be bothered by it due to a change in perception of tinnitus. There are several components in tinnitus retraining therapy that helps complete habituation of the noise. One of the most important components is counseling, which involves education on tinnitus and its effects. This step is important because it addresses the patient's questions and concerns regarding tinnitus, and through this, the patients who are struggling to cope with tinnitus will be less anxious about the condition. Another similar and important component of retraining therapy is psychological therapy. Psychological therapy teaches the patient to ignore the tinnitus noise and helps manage stress levels. This therapy allows the patient to perceive that tinnitus is no longer a source of danger. Lastly, another important component of retraining therapy is wearing a device behind the ear that generates broad-band noise. This device will allow the patient to divert their attention away from the tinnitus.

The goals of the clinical information system is to support transactions in a clinic that treats patients who have hearing disorders using tinnitus retraining therapy. The transactions that are supported by the system include registering a new patient, viewing and editing added patients, adding a new visit, viewing and editing existing visits, assigning a category to a patient, performing audiological evaluation, registering medical history, and finally scheduling the next visit.

The first transaction, registering a new patient, involves many steps. To register a patient into the system, it needs to enter the patient's identification number, first and last name of the patient, date of birth, phone number, gender, address, social security number, insurance number. The optional information to register a patient include the patient's occupation, current work status, and education degree. Lastly, the system will also enter the patient's etiology of both tinnitus and hyperacusis and medication information to finish registration of the patient. Once the patient is added to the system, the system is able to view and edit the patient's information, or delete the most recently added system.

The next transaction, adding a new visit, also involves many steps. In order for the system to add a visit for a patient, the patient must first be registered in the system. Once the patient is already registered, the patient can then navigate to the patient database and add a visit by first entering a new visit. The information that needs to be entered to make a new visit are date of visit, sequence number of the visit, treatment progress, treatment methods, and tinnitus/hyperacusis interview form. Then, another set of information that needs to be added to the system are audiological evaluation results. The results include recording loudness discomfort levels for the left and right ear, tinnitus pitch, tinnitus match and match type, hearing threshold, minimum masking levels for the left and right ear, pure tone audiogram for the left and right ear, and additional comments. Once the audiological evaluation has been entered, the system also requires the user to assign a category for the patient. Once these three components have been completed, the system then finishes adding a new visit for the patient. Visit history can only be accessed when the patient has completed the registration and has had at least one visit.

Lastly, scheduling the next visit can be done by choosing a patient and entering the date and time for the next visit. Additionally, if at any point there is a need to remove a patient from the system, the system can remove a patient by going to the list of patients and deleting the patient from the list.

References

Sandhya, P. (2019, March 05). Tinnitus. Retrieved May 20, 2020, from

<https://www.mayoclinic.org/diseases-conditions/tinnitus/symptoms-causes/syc-20350156>

Tinnitus Retraining Therapy - Facts about Tinnitus Retraining Therapy. (n.d.). Retrieved May 21,

2020, from <https://www.hear-it.org/Tinnitus-Retraining-Therapy>

Design

Use Cases:

Bold - This step is defined as a use case

Steps with .1 means variation.

Register New Patient

1. System opens up on Page 1 of the UI
2. Click
 - 1.1 "Register New Patient"
 - i. System changes UI to page 2
 - 1.2 Click "View Existing Patients"
 - i. System changes UI to page 7
 - ii. Click "Add new"
 - aa. System changes UI to page 2
3. Input required information
 - a. ID number(ordering number)
 - b. Date Added
 - c. First Name
 - d. Last Name
 - e. Birthday
 - f. Gender
 - g. Phone number
 - h. Address
 - i. Street Address 1
 - ii. Street Address 2
 - iii. City
 - iv. State
 - v. Zip Code
 - vi. Country
 - i. SSN
 - j. Insurance Number

4. Click "Next"
 - 3.1 Some required information is missing or invalid: prompt again for information
 - 3.2 System changes UI to Page 3
5. Input optional information
 - a. Occupation
 - b. Work status
 - c. Education degree
6. Click
 - 1.1 Click "back"
 - aa. System changes UI to page 2
 - 1.2 Click "next"
 - aa. System changes UI to Page 4
7. Input Medical History
 - a. Tinnitus and hyperacusis status
 - i. Etiology (cause of disease) with description
 - 1.1 Click "back"
 - aa. System changes UI to page 3
 - 1.2 Click "next"
 - aa. System changes UI to page 5
8. Pharmacology
 - i. current medication
 - ii. medicament's name
 - iii. generic dose
 - iv. Duration
 - v. chemical category
 - vi. Action
 - vii. Application
 - viii. usual dose
 - ix. whether the medication induced tinnitus as a side effect.
 - 1.1 Click "Add Medication" until all medications taking are included
 - 7.1 Some fields are incomplete: prompt again for completion
 - 1.2 Click "Finish"
- 9. Add Patient to Database**
10. System changes UI to page 6

Cancel Registration of New Patient

- 1. Register New Patient**
2. Click "undo"

3. Delete Patient from Database

View existing patients

1. Register new patient (need a patient to view one)
2. Click "View existing patients"
- 3. Query Patients from Database**

Edit existing patient's info

- 1. View existing patients**
2. Select a patient
3. Click "info"
 - a. User modifies information
 - i. Click "update"
- 4. Modify Patient from Database**

Add new visit for existing patient

- 1. View existing patients**
2. Click on a specific patient
3. Click "Add Visit"
4. Input date
5. Input visit sequence number (positive integer)
6. (Required if sequence number 0) input Tinnitus/Hyperacusis Initial Interview Form
7. Input treatment progress
8. Input treatment methods applied
 - a. Sound therapy
 - b. Real-ear measurements
 - c. Counseling
9. Click "Next"
 - a. System changes UI to page 12
- 10. Add new audiological evaluation**
- 11. Add Visit to Database**

Add new audiological evaluation

- 1. Add New Visit for Existing Patient**

2. Input result of audiological evaluation

- a. pure-tone audiogram for the left and right ear in all frequencies (from 0.25 kHz to 12kHz)
- b. Loudness discomfort levels (LDL) for the left and right ear in frequencies 0.5 kHz to 12 kHz
- c. Tinnitus
 - i. Pitch
 - ii. Match
 - iii. match type
- d. thresholds of hearing
- e. minimal masking levels for the left and right ear
- f. comments

3. Click “Next”

11.1 Some fields are empty: prompt for completion

4. Assign category to patient

- g. View patient medical history
- h. View current medication taking
- i. Assign category (0-4)
 - i. 0 - tinnitus present but no impact
 - ii. 1 - tinnitus w/ high impact
 - iii. 2 - hearing problem present & relevant
 - iv. 3 - hyperacusis is a major problem
 - v. 4 - prolonged tinnitus exacerbation

c.1. Invalid category: Prompt for valid category number

5. Click “Finish”

5.1 Some fields are empty: prompt for completion

5.2 System changes UI to page 7

6. Add Audiological Evaluation to Database

View existing visit

1. View existing patients

2. Select a patient

3. Add new visit for existing patient (need at least 1 to view them)

4. Click on “Visit History”

5. Query Visit from Database

6. System changes UI to page 14

Edit existing visit

- 1. View existing visits**
2. Select Visit
3. Click "Info"
4. Input changes
5. Click "Edit Information"
- 6. Modify Visit from Database**

Schedule next visit

- 1. View existing patient**
2. Select a patient
- 3. Click "Schedule Visit"**
4. System opens popup
5. Input date and time range of next visit
6. Click "Enter"
 - a. Popup closes
 - b. Add Scheduled Visit to Database**

-----System Side Use Case-----

Add Patient to Database

- 1. Register New Patient**
2. Add the patient to the PatientModel

Delete Patient from Database

- 1. Cancel Registration of New Patient**
2. Delete the patient from the PatientModel

Add Visit to Database

- 1. Add new visit for existing patient**
2. Adds visit to the Patient in the PatientModel

Add Scheduled Visit to Database

- 1. Schedule next Visit**
2. Adds scheduled visit to the Patient in the PatientModel

Add Audiological Evaluation to Database

- 1. Add Visit to Database**
2. Add evaluation to the Patient in the PatientModel

Query Patients from Database

1. **View existing patients**
2. Query list of Patient from PatientModel

Query Visits from Database

1. **View existing visits**
2. Query list of Visit from Patient in PatientModel

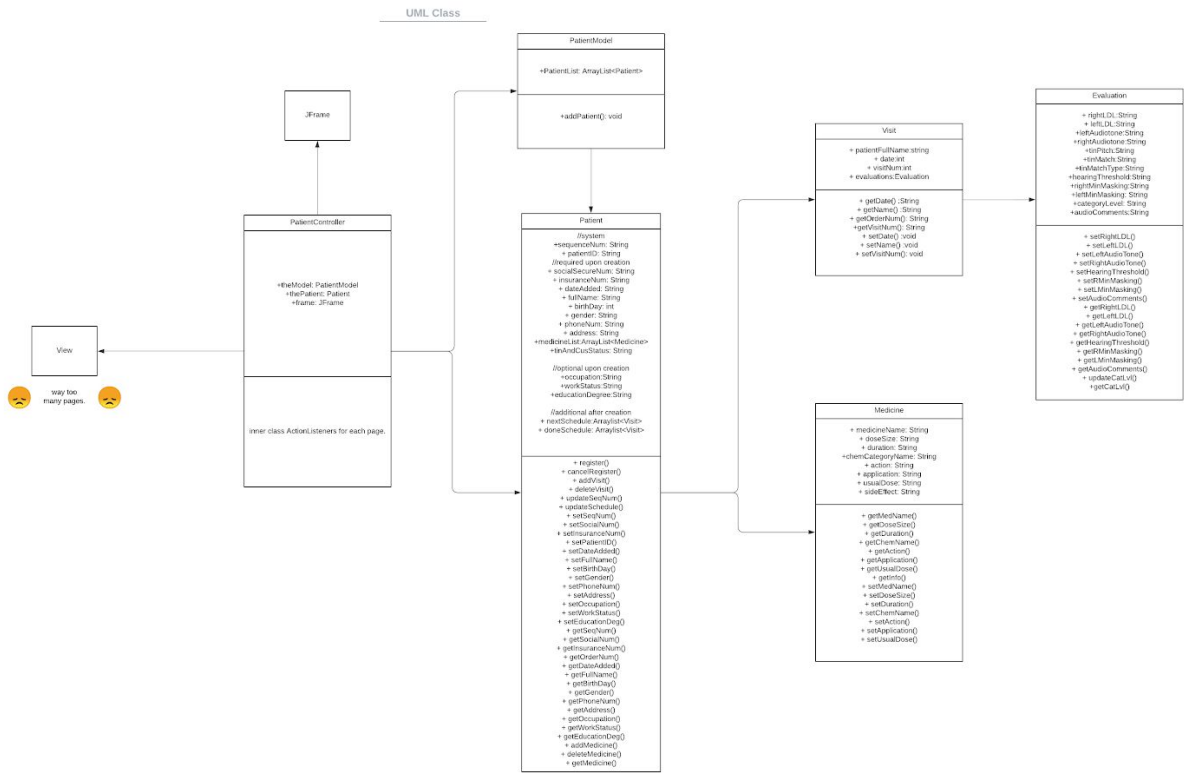
Modify Patients from Database

1. **Edit existing Patients**
2. Alter correct patient from PatientModel

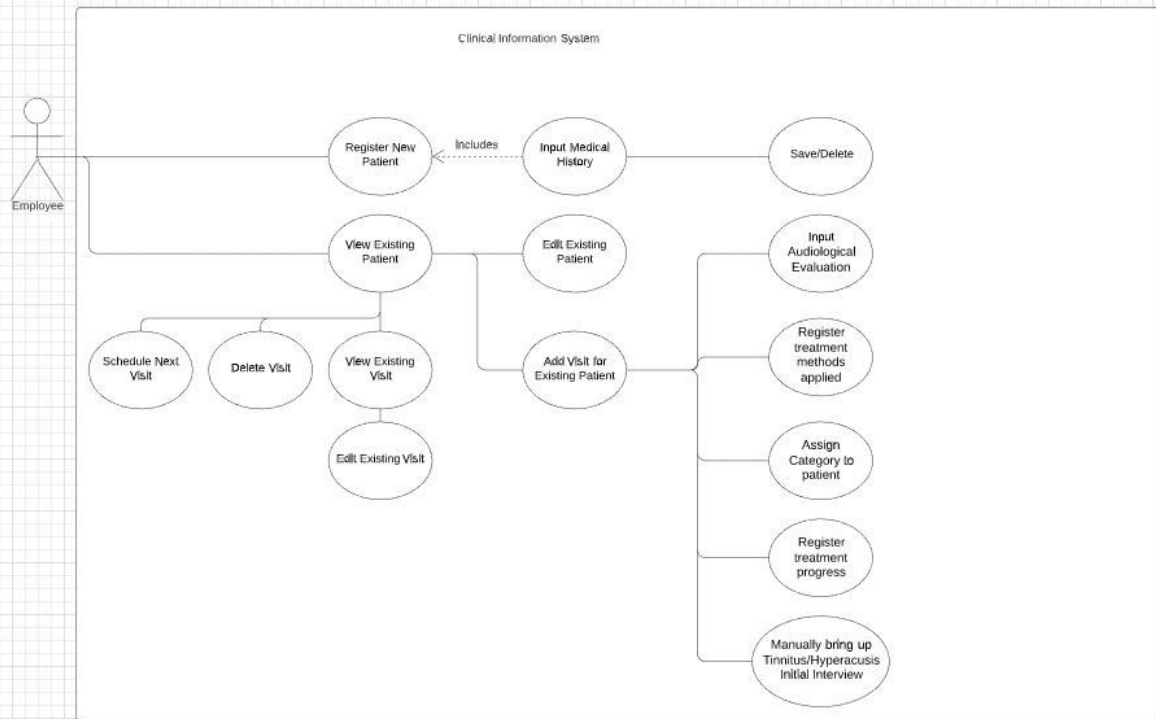
Modify Visits from Database

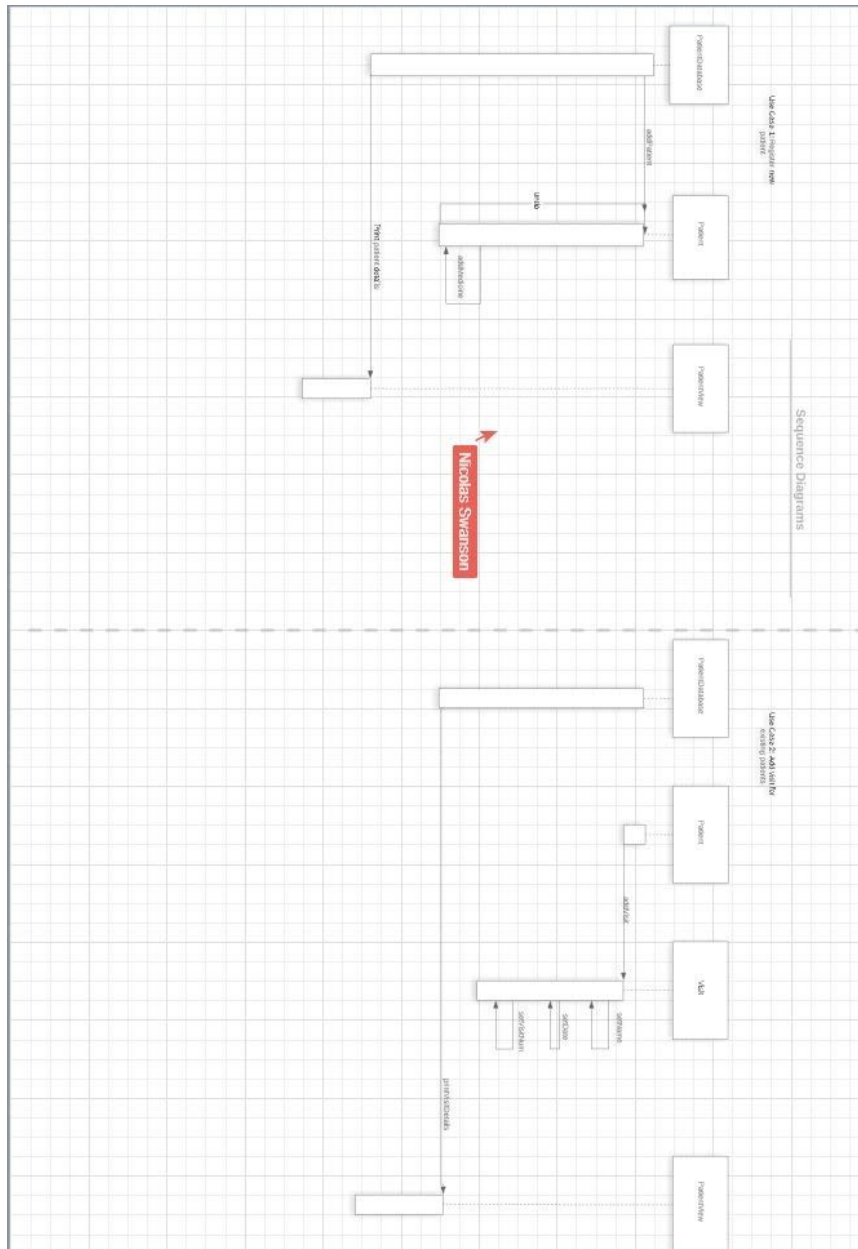
1. **Edit existing Visits**
3. Alter correct visit from Visit for the Patient in PatientModel

Diagrams



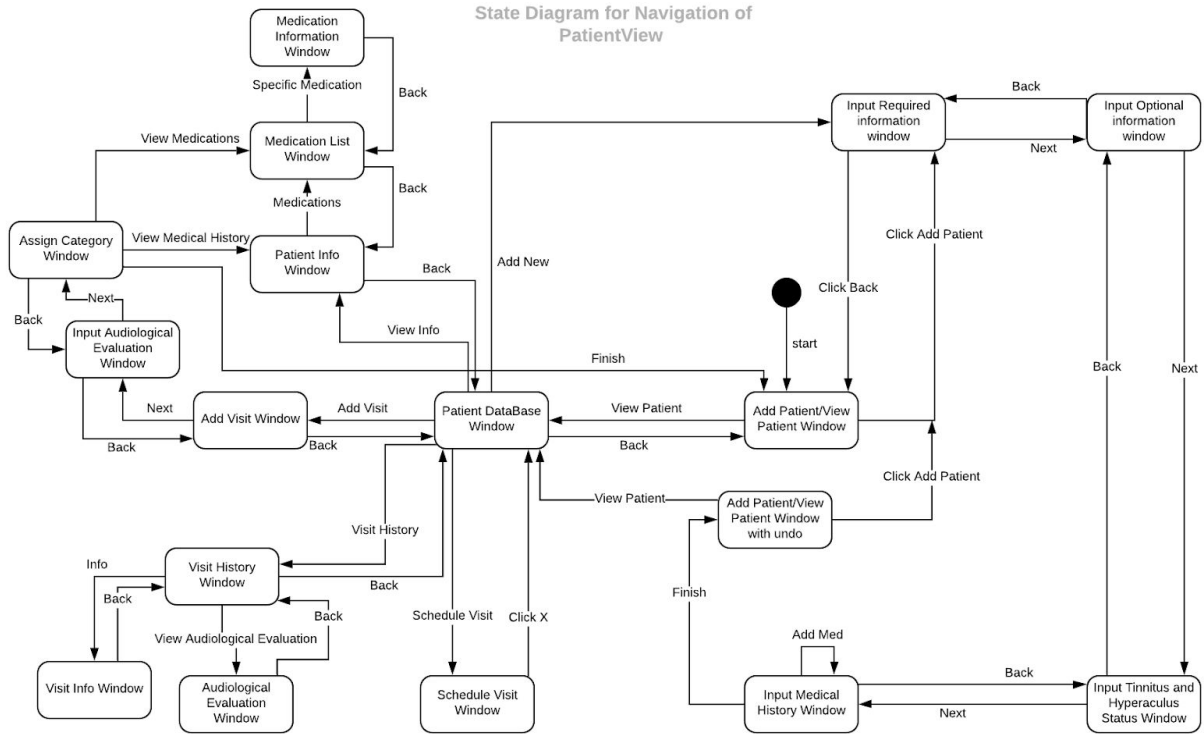
Use Case Diagram





Note: the upper image contains two sequence diagrams.

State Diagram for Navigation of PatientView



UI Design

Note: These designs are best case scenarios. If it is too hard to implement, we will simplify it.

Implementation Changes: Highlighted in Yellow

1



This is a frame with 2 big panels. In each panel, there will be a button corresponding with the text. ActionListener on Register New Patient Button which will bring up image 2. ActionListener on View Existing Patients which will bring up image 7. I chose this design because these are the two main actions, and it is very straightforward and simple.

Patient Required Information

Required Registration Information
All fields must be completed correctly

ID Number	<input type="text"/>	Street Address 1	<input type="text"/>
Date Added	<input type="text"/>	Street Address 2	<input type="text"/>
First Name	<input type="text"/>	City	<input type="text"/>
Last Name	<input type="text"/>	State	<input type="text"/>
Birthday	<input type="text"/>	Zip Code	<input type="text"/>
Gender	<input type="text"/>	Country	<input type="text"/>
Phone Number	<input type="text"/>	Insurance Number	<input type="text"/>
SSN	<input type="text"/>		

This is a borderlayout with buttons on the bottom, text field on top, and two panels in the center. The two center panels will have two panels with box layout. One of the panels will have text displays and the other will have spaces the user can input text.

Back button will go back to image 1.

Next button will go to image 3.

Most screenshots follow the same theme, so the criteria will be explained here.

Intuitiveness: Each text input field is described by a text description of what should be in input. Each button is labeled with an icon for instant understanding. The buttons are placed intuitively. For example, “Back” is on the left side of the page while “Next” is on the right side of the page.

Navigability: The “Back” and “Next” buttons offer easy to access navigation between pages. The title at the top ensures users always know what the objective is.

Graphics: The graphics follow a minimalist structure as described by “10 Usability Heuristics for User Interface Design” by Jacob Nielsen. The blue is a common hospital color to reinforce the professional nature and atmosphere of the clinic. The two lines at the bottom offer a variation to an otherwise stale white background. Buttons are always in blue to pop out for the user.

Usability: Text inputs are easy to access and are presented in an orderly fashion.

User-friendliness: Information appears in intuitive order. Instructions for use of the system are clearly defined and labeled. All information presented is relevant.

3

The screenshot shows a web browser window with a dark grey title bar containing three small circular icons on the left and the text 'New Patient's Optional Info' on the right. Below the title bar is a blue header bar with the text 'Optional Information' in white, and 'Fields are not mandatory' in a smaller white font below it. The main content area is white and contains three labels on the left: 'Occupation', 'Work Status', and 'Education Degree'. To the right of each label is a white rectangular text input field. At the bottom of the form are two blue buttons: '← Back' on the left and '→ Next' on the right. Below the buttons are two horizontal blue bars of increasing width, creating a footer-like effect.

This uses the same layout as image 2 except with only 1 panel in the middle. Back goes to image 2. Next goes to image 4.

The screenshot shows a web-based medical history form. At the top, a dark grey header bar contains the text 'Medical History: Tinnitus and Hyperacusis Status'. Below this is a blue header bar with the title 'Tinnitus and Hyperacusis Status' and the subtitle 'Describe in detail'. The main content area is white and contains the label 'Etiology Description:' followed by a large, empty text input field. The input field contains faint, wavy lines suggesting a placeholder or a very light watermark. At the bottom of the form, there are two blue buttons: 'Back' on the left and 'Next' on the right. The entire form is framed by a thin blue border.

Also similar to image 2, except with only 1 text field and a huge text input field in the middle.

Very intuitive as all you have to do is fill out the blank area in the middle then press "Next."

5

The screenshot shows a web browser window with the title 'Medical History: Medication'. The page has a blue header with the text 'Medical History' and 'Complete fields in detail'. On the right side of the header is a blue button labeled 'FINISH'. Below the header, there are two columns of input fields. The left column contains: 'Current Medication', 'Medicant's Name', 'Generic Dose', 'Duration', and 'Chemical Category'. The right column contains: 'Action', 'Application', 'Usual Dose', and 'Side Effects'. At the bottom left is a blue button labeled '← Back', and at the bottom right is a blue button labeled '→ Add Med'.

Similar layout to image 2, but with a “Finish” button at the top right. This button goes to image 6. Users can continuously add medications until they press finish. It is intuitive to press “Add Med” to add medications and to press “Finish” when the user is finished with adding medications.

Switched placement of finish button and add med button.

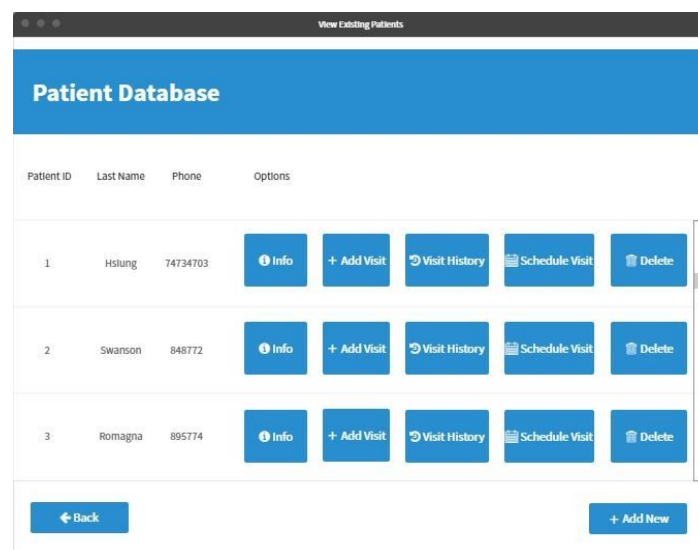
6

The screenshot shows a web browser window with the title 'Clinic Employee Page'. The page has a blue background and is divided into two main sections. The left section is labeled 'Register New Patient' and the right section is labeled 'View Existing Patients'. In the bottom left corner, there is a white 'Undo Panel' with the text: 'You have just added a new patient. Click "Undo" to undo.' Below this text is a button labeled 'Undo'. At the very bottom of the panel, it says 'This panel will automatically disappear in one minute.'

This is the same layout as image 1 except with another panel at the bottom left with a text field and “Undo” button. The undo button cancels the add transaction that just happened. This adds to user-friendliness as they can undo their mistake of adding in a patient.

Undo button is no longer a pop up. It is just a button.

7



This is the list of all the patients. The overall layout is border. In the center, there will be panels for each row. In each row there will be text fields and buttons in flow layout. Users can scroll through the patients to look at more patients.

Buttons:

Info - goes to image 8

Add Visit- goes to image 10

Visit History - goes to image 14

Schedule Visit - goes to image 15

Delete- deletes this patient

Back- goes back to image 1

Add new - goes to image 2

The reason why this is not the default page even though it has all the functionalities is to preserve the privacy of patients. It seemed intrusive to have the default page to be the list of all patients, so the current default page is used to hide the list of all patients. This UI is easy to use with many buttons explaining exactly what it does.

Intuitiveness - With ID, Last Name, and Phone Number displayed, it is easy to read the identifying aspects of a patient. It is intuitive that buttons on the same row of a patient correspond to that patient.

Navigability - The buttons provide easy access to each functionality.

Graphics- The blue buttons focus the user's attention and follow minimalism.

Usability- It is easy to use and provides the necessary functionalities.

User Friendliness - Everything is easy to see and use. If the wrong button is pressed, users can always press “Back.”

Instead of a series of buttons for each patient, user simply selects a patient and presses the corresponding button.

8

The screenshot shows a web interface titled "Patient Information". Below the title is a blue header bar with the text "Information of Patient Hsiung". Underneath is a table with 10 columns: Patient ID, Last Name, First Name, Date Added, Birthday, Gender, Phone Number, Address, SSN, and Insurance Number. The table contains one row of data for Patient ID 1, Last Name Hsiung, First Name Johnson, Date Added 4/20/2020, Birthday 6/5/2000, Gender Male, Phone Number 6503803805, Address 1481 Oliver Place, SSN xxx-xx-xxxx, and Insurance Number xxxxxxxx. To the right of the table is a link "Edit Information". Below the table is a section with the label "Etiology:" and a large text input field. At the bottom left is a blue button with a left arrow and the text "Back". At the bottom right is a blue button with a magnifying glass icon and the text "Medications".

Patient ID	Last Name	First Name	Date Added	Birthday	Gender	Phone Number	Address	SSN	Insurance Number
1	Hsiung	Johnson	4/20/2020	6/5/2000	Male	6503803805	1481 Oliver Place	xxx-xx-xxxx	xxxxxxxx

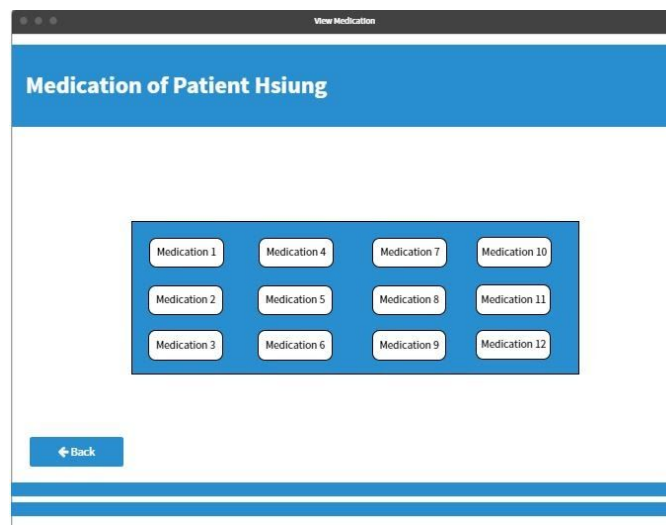
Etiology:

[Edit Information](#)

[Back](#) [Medications](#)

This image is for viewing the information of a specific patient. This will be a borderlayout with border layout at the bottom. The center is a table with the information of a specific patient. The bottom is a text field with the description of etiology in the center. Users can press “Edit Information” to change the information. Users can press “Medications” to go to image 9 or “back” to go back to image 7.

This design is easy to read with all the information presented in an orderly way. The buttons are intuitive and its actions are also intuitive.



This will be a borderlayout with grid layout in the center. The center will contain a panel with buttons for each medication of this particular patient. Users can press any medication button to go to image 10. The user can easily access the information for any specific medication. “Back” will go back to image 8.

The screenshot shows a window titled "Patient Information". Below the title bar is a blue header with the text "Patient Hsiung's Medication 1 Information". Below the header is a table with 8 columns: Medication Name, Generic Dose, Duration, Chemical Category, Action, Application, Usual Dose, and Side Effects. The table contains one row of data: Medication 1, 1, 10 seconds, xxxxx, xxxxx, xxxxx, xxxxx, and No side effects. Below the table is a link "Edit Information" and a "Back" button.

Medication Name	Generic Dose	Duration	Chemical Category	Action	Application	Usual Dose	Side Effects
Medication 1	1	10 seconds	xxxxx	xxxxx,xxxxx	xxxxx	xxxxxx	No side effects

[Edit Information](#)

[Back](#)

This could be a box layout or border layout. It is similar to image 8. Information is concise and easy to read. "Edit Information" is small enough to not be distracting but also big enough to be easily found when searching for it. Button "Back" goes to image 9.

11

The screenshot shows a window titled "Patient Required Information". Below the title bar is a blue header with the text "New Visit Information for Patient Hsiung" and "All fields must be completed correctly". Below the header is a form with several fields: Date of Visit, Sequence Number, Treatment Progress, and Treatment Methods. The Treatment Methods section has three checkboxes: Sound Therapy, Real Ear Measurement, and Counseling, all of which are checked. To the right of these fields is a large text area labeled "Tinnitus/Hyperacusis Interview Form". Below this text area is a note "Required for Sequence Number 0". At the bottom of the form are "Back" and "Next" buttons.

Date of Visit:

Sequence Number:

Treatment Progress:

Treatment Methods:

- ☒ Sound Therapy
- ☒ Real Ear Measurement
- ☒ Counseling

Tinnitus/Hyperacusis Interview Form

Required for Sequence Number 0

[Back](#) [Next](#)

This is for adding a visit to a specific patient. It's layout is similar to image 2 except for the Tinnitus/Hyperacusis Interview Form located in the right panel of the center, it is just one panel with a text description and text input field. The checkboxes of treatment

methods can also just be a text input field. “Back” goes back to image 7. “Next” goes to image 12.

12

Patient Required Information

Results of Audiological Evaluation for Patient Hsiung

Loudness Discomfort Level Left Right

Tinnitus Pitch

Match

Match Type

Hearing Threshold

Minimum Masking Level Left Right

Additional Comments

Pure Tone Audiograph for Left and Right Ear

Hearing Level in Decibels (dB)

Frequency in Hertz (Hz)

← Back

→ Next

This will be a border layout for the whole frame. In the center of the border layout, there will be 2 panels. The first panel on the left will also be a border layout containing a variety of text fields and text input fields. The second panel will also be a border layout. The center of it is where the user can input the audiograph and the text fields are on top, left, and bottom. Text input fields are easy to see and fill out.

Assign category now appears under addition comments.

13

Assign Category

Assign Category for Patient Hsiung

Check Medical History and Medication

- ☒ 0 - tinnitus present but no impact
- ☐ 1 - tinnitus w/ high impact
- ☐ 2 - hearing problem present & relevant
- ☐ 3 - hyperacusis is a major problem
- ☐ 4 - prolonged tinnitus exacerbation

[View Medical History](#)

[View Medications](#)

[← Back](#) [→ Finish](#)

This layout reuses the same layout as image 2. The difference is the right panel in the center will be a box layout with the two buttons. The left panel in the center could just be a text input field with the same description.

The two buttons “View Medical History” and “View Medications” are big to highlight the fact that the user is supposed to view the information first before assigning a category.

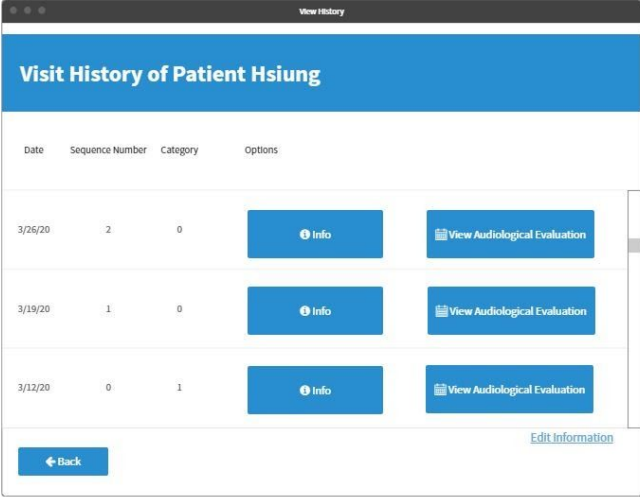
“View Medical History Button” goes to image 8.

“View Medications” goes to image 9.

“Back” goes to image 12.

“Finish” goes to image 7.

14



Date	Sequence Number	Category	Options
3/26/20	2	0	<button>Info</button> <button>View Audiological Evaluation</button>
3/19/20	1	0	<button>Info</button> <button>View Audiological Evaluation</button>
3/12/20	0	1	<button>Info</button> <button>View Audiological Evaluation</button>

Back [Edit Information](#)

The format is similar to image 7 (patient database). This is the visit history of a particular patient. It presents date, sequence number, and category to help identify which visit the

user is looking for. Users can press “info” to see the visit in detail (image 15). “Back” goes back to image 7.

User selects a visit and presses corresponding button.

15

The screenshot shows a web form titled "Visit 0 of Patient Hsiung". It contains several input fields: "Date of Visit" and "Sequence Number" are text boxes with placeholder text "asdf"; "Treatment Progress" is a larger text box with placeholder text "asdfasdf"; "Treatment Methods" includes three checked checkboxes: "Sound Therapy", "Real Ear Measurement", and "Counseling". To the right is a large empty box labeled "Tinnitus/Hyperacusis Interview Form". At the bottom left is a blue "Back" button with a left arrow, and at the bottom right is a blue "Edit Information" link.

This format is similar to image 11 (input new visit). Users can press “Edit Information” to edit information. Users can press “Back” to go back to image 14.

16

The screenshot shows a web form titled "Results of Audiological Evaluation for Patient Hsiung". It contains several input fields: "Loudness Discomfort Level" has "Left" and "Right" text boxes with placeholder text "asdf"; "Tinnitus" includes "Pitch", "Match", and "Match Type" text boxes with placeholder text "asdf"; "Hearing Threshold" is a text box with placeholder text "asdf"; "Minimum Masking Level" has "Left" and "Right" text boxes with placeholder text "asdf"; and "Addition Comments" is a large text box with placeholder text "asdf". To the right is a large empty box labeled "Pure Tone Audiograph for Left and Right Ear" with a vertical label "Hearing Level in Decibels (dB)" on the left and "Frequency in Hertz (Hz)" at the bottom. At the bottom left is a blue "Back" button with a left arrow, and at the bottom right is a blue "Edit Information" link.

This format is similar to image 12 (input result of audiological evaluation). Users can press “Edit Information” to edit information. Users can press “Back” to go back to image 14.

17

The screenshot shows a web application titled "View Existing Patients" with a "Patient Database" section. It contains a table with three patients and a modal for scheduling a visit.

Patient ID	Last Name	Phone	Options
1	Hsiung	74734703	<div>Info</div> <div>+ Add Visit</div> <div>Visit History</div> <div>Schedule Visit</div> <div>Delete</div>
2	Swanson	848772	<div>Info</div> <div>+ Add Visit</div> <div>Visit History</div> <div>Schedule Visit</div> <div>Delete</div>
3	Romagna	895774	<div>Info</div> <div>+ Add Visit</div> <div>Visit History</div> <div>Schedule Visit</div> <div>Delete</div>

Modal: Schedule Visit for Patient Hsiung

Date:

Time Start:

Time End:

Buttons: X, y, + Add Visit, Visit History, Schedule Visit, Delete

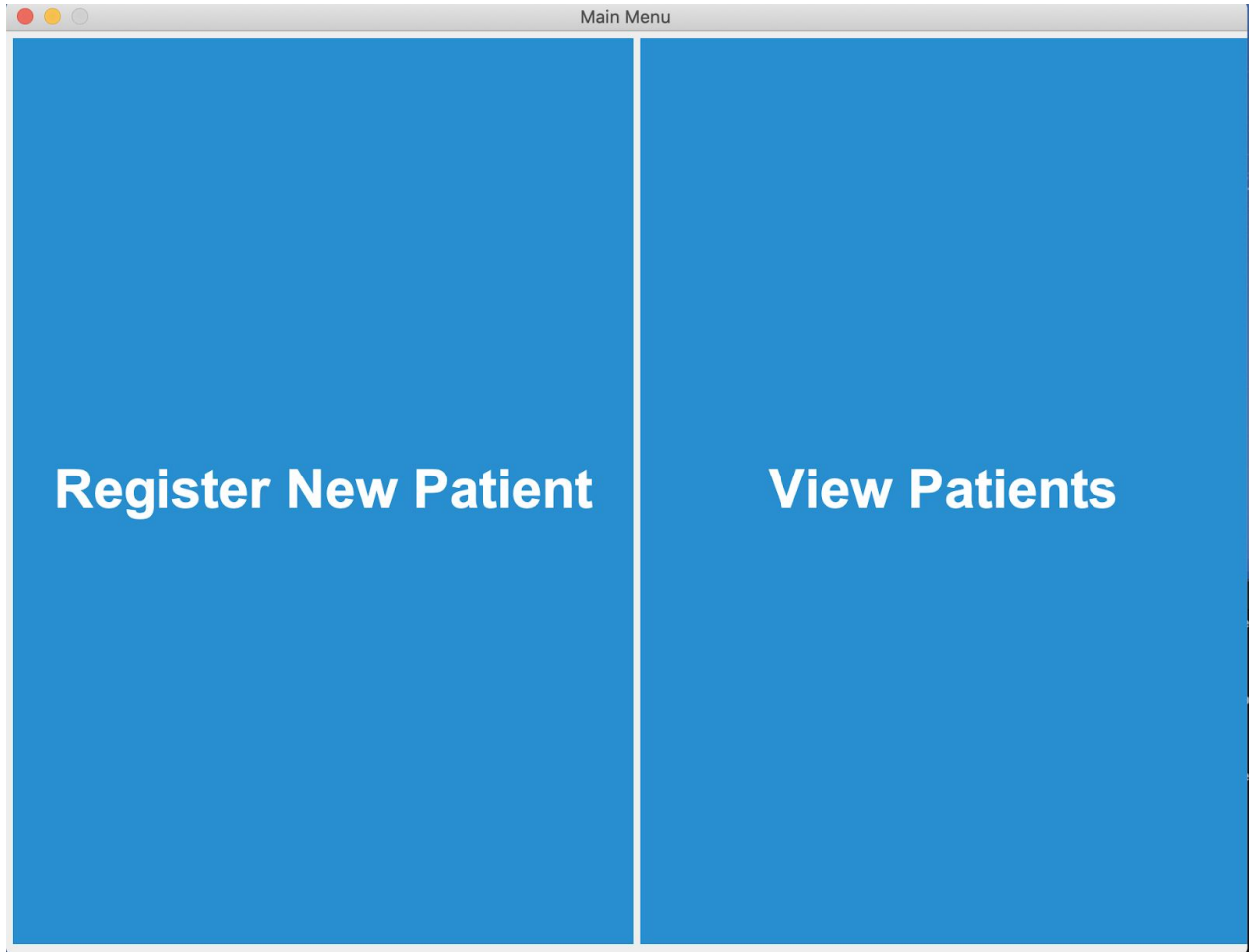
Footer: Back, + Add New

This is for when the user presses “Schedule Visit.” A panel will pop up in the center. It will have text descriptions and text input fields. The user can press “Enter” on their keyboard to schedule this visit or the “x” in the top left to exit.

This is efficient to use as there is no need for an extra page to schedule a visit.

Added “Enter” button to pop up.

Implementation



Patient Registration

Required Registration Information

ID Number	1234567	Street Address 1	1481 North oak Ridge
Date Added	06/05/2013	Street Address 2	1234 nono yes
First Name	Johnson	City	Santa Clara
Last Name	Hsiung	State	CA
Birthday	06/05/2000	Zip Code	91023
Gender	Male	Country	USA
Phone Number	6503803804	Insurance Number	12376869
Social Security Number	99999999999		

Back

Next

Optional Information

Optional Registration Information

Occupation	plumber
Work Status	employed
Education Degree	high school

Back

Next

Tinnitus and Hyperacusis Status

Optional Registration Information

Etiology Description

Has lots of pains and stuff.

Back

Next

Medical History

Current Medication

pepsi

Medicant's Name

Johnson

Generic Dose

50 mg

Duration

25 seconds

Chemical Category

gas

Action

injection

Application

needle

Usual Dose

1230 mg

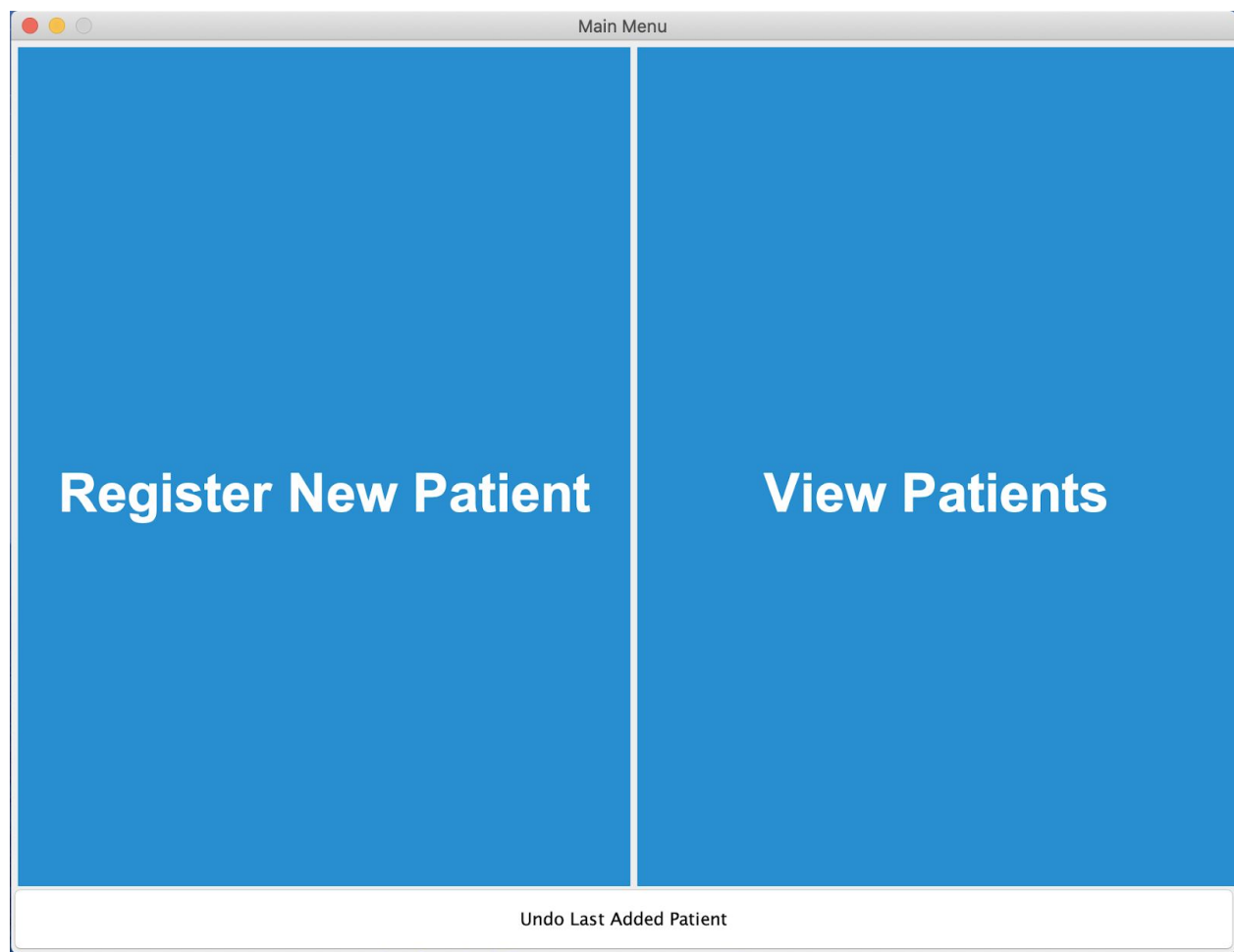
Side Effects

none

Add Med

Back

Finish



Patient Database

Patient Database

Select a Patient

01309123

JohnsonHsiung

000001

MarcoRomagna ;)

1234567

Johnson Hsiung

Info

+ Add Visit

Visit History

Schedule Visit

Delete

Back

Add New

Patient Information

Information of Hsiung

ID	Date Added	FirstName	LastName	Birthday	Gender	PhoneNumber	SSN	Address1
1234567	06/05/2013	Johnson	Hsiung	06/05/2000	Male	6503803804	99999999999	1481 North o...
Address2	City	State	Zip	Country	Insurance #	Occupation	WorkStatus	Education
1234 nono yes	Santa Clara	CA	91023	USA	12376869	plumber	employed	high school

Update Inputted Info of Table and Description

Etiology

Has lots of pains and stuff.

Back

Medication

Medication of Patient Hsiung

pepsi

Back

Medication Information

Information of pepsi for Johnson

Current Medication	Medicant's Name	Generic Dose	Duration	Chemical Category
pepsi	Johnson	50 mg	25 seconds	gas
Action	Application	Usual Dose	Side Effects	
injection	needle	1230 mg	none	

Update Inputted Info of Table

Back

Visit Information Page

New Visit Information for Patient Hsiung

Date of Visit	06/05/2014	Tinnitus/Hyperacusis Interview Form Good
Sequence Number	1	
Treatment Progress	fast	
Treatment Methods	<div><div><input checked="" type="checkbox"/> Sound Therapy</div><div><input type="checkbox"/> Real Ear Measurement</div><div><input checked="" type="checkbox"/> Counseling</div></div>	
<div>BackNext</div>		

Audiological Information

Results of Audiological Evaluation for Patient Hs

Loudness Discomfort Level

Left50Right30

Tinnitus

Pitch2

match1

Match Type34

Hearing Threshold

1230

Minimum Masking Level

Left123Right123

Additional Comments

Great

Pure Tone Audiograph for Left & Right Ear

Back

Next

Assign Category for Patient Hsiung

☐ 0 - tinnitus present but no impact

☒ 1 - tinnitus w/ high impact

☐ 2 - hearing problem present & relevant

☐ 3 - hyperacusis is a major problem

☐ 4 - prolonged tinnitus exacerbation

Medications

Medical History

Back

Finish

Patient Visit History

Visit History of Hsiung, Johnson

Select a Visit

06/05/20141

Info

View Evaluation

Back

Visit 1 of Patient Hsiung

		Tinnitus/Hyperacusis Interview Form
		Good
Date of Visit	06/05/2014	
Sequence Number	1	
Treatment Progress	fast	
Treatment Methods	<input checked="" type="checkbox"/> Sound Therapy	
	<input type="checkbox"/> Real Ear Measurement	
	<input checked="" type="checkbox"/> Counseling	
Back		Edit Infor...

Audiological Information

Results of Audiological Evaluation for Patient Hs

Loudness Discomfort Level

Left50Right30

Tinnitus

Pitch2

match1

Match Type34

Hearing Threshold

1230

Minimum Masking Level

Left123Right123

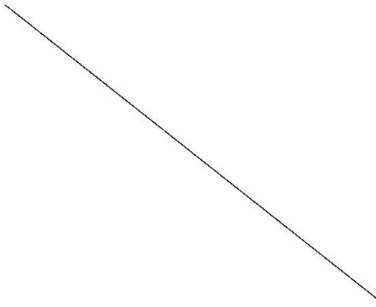
Additional Comments

great

Category

3 – hypercausis is a major problem

Pure Tone Audiograph for Left & Right Ear



Back

Edit Infor...

Patient Database

Patient Database

Select a Patient

01309123 JohnsonHsiung

000001

1234567

Info

Visit

ory

Visit

Schedule a Visit

Date:

Start Time:

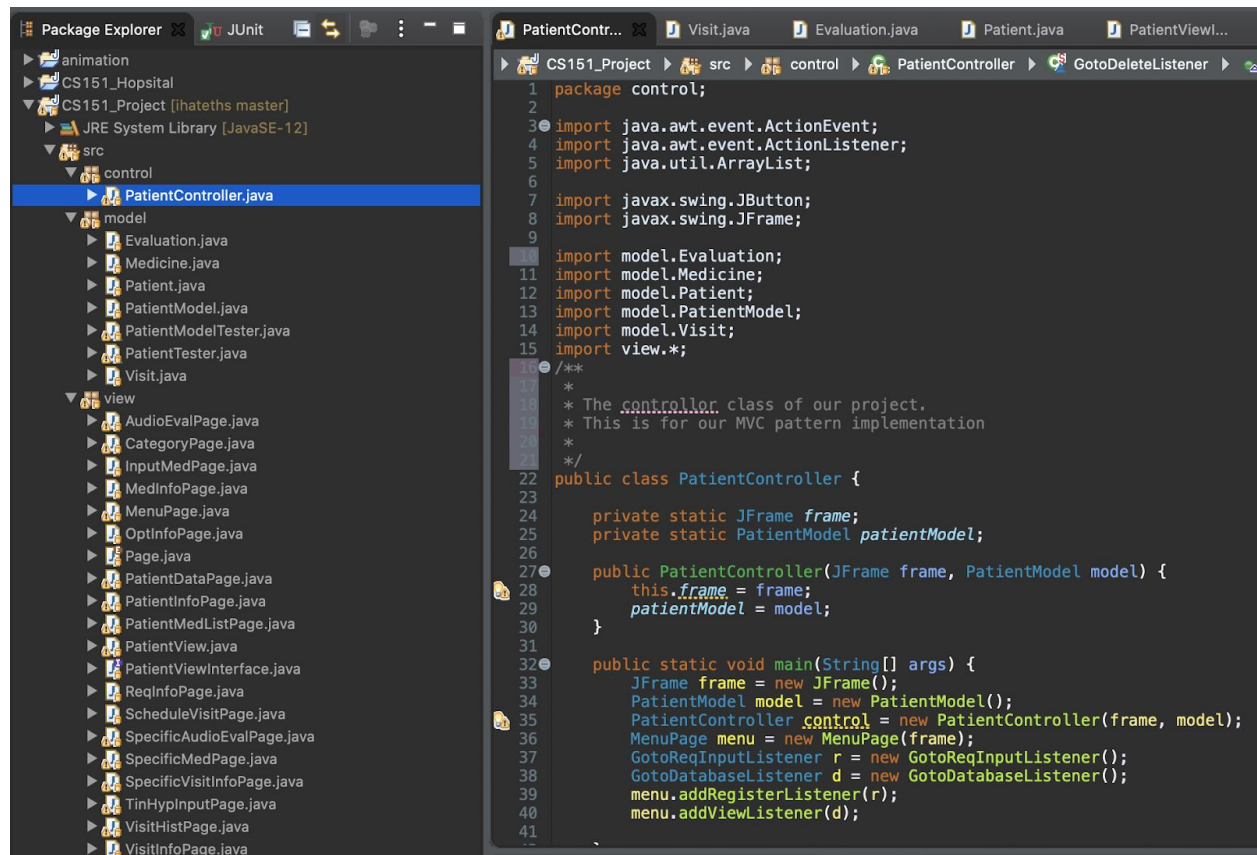
End Time:

Enter

Back

Add New

Description of Implementation



The application of the MVC pattern can be seen in this screenshot. In the project directory, each part of the MVC pattern is in a separate package. The control has access to the model and view package while the model and view do not access any other package. The PatientController takes in both the frame and the PatientModel. The frame is to display all the pages in the View package on the same frame. The point of the PatientModel is to keep track of the patients in the database. The main method creates a frame, a patient model, a menu page, and the corresponding action listeners for each navigation button. The ActionListeners in the Controller take care of moving in between the pages in the View package.

```
if (db.isPatientSelected() == true) // then button will work
{
    Patient patient = db.whichPatient();
    PatientInfoPage infoPage = new PatientInfoPage(frame, patient.getInfo());
    GotoDatabaseListener GotoData = new GotoDatabaseListener();
    GotoMedicationListener GotoMedication = new GotoMedicationListener(patient);
    infoPage.addBackListener(GotoData);
    infoPage.addNextListener(GotoMedication);

    infoPage.addUpdateListener((a) -> {
        String[] info = infoPage.getEditedInfo();
    });
}
```

In this screenshot, it can be seen that PatientInfoPage (View Package) does not take in Model-only parameters. The controller extracts the necessary information from the model and inputs it into the PatientInfoPage. This is to keep View and Model completely separate. This design is consistently used whenever information from the Model needs to be displayed.

```

1 package view;
2
3 import java.awt.BorderLayout;
12
13 public class MenuPage {
14     private JButton registerButton;
15     private JButton viewButton;
16     private JFrame frame;
17     private JButton undoButton;
18     boolean isUndo;
19     public MenuPage(JFrame frame)
20     {
21         this.frame = frame;
22         frame.setResizable(false);
23         frame.getContentPane().removeAll();
24         registerButton = new JButton("Register New Patient");
25         viewButton = new JButton("View Patients");
26         frame.setTitle("Main Menu");
27         frame.setSize(910, 700);
28         JPanel register = new JPanel();
29         register.setSize(455, 700);
30         frame.add(register, BorderLayout.LINE_START);
31
32
33         registerButton.setPreferredSize(new Dimension(455, 665));
34         registerButton.setOpaque(true);
35         registerButton.setBackground(new Color(41, 142, 208));
36         registerButton.setForeground(Color.white);
37         registerButton.setBorderPainted(false);
38         registerButton.setFont(new Font("Arial", Font.BOLD, 40));
39         register.add(registerButton);
40
41
42
43
44         JPanel view = new JPanel();
45         view.setOpaque(true);
46
47         view.setSize(455, 700);
48         frame.add(view, BorderLayout.CENTER);
49

```

This is the code for the MenuPage; Most View classes are done this way. The constructor makes the page on the given frame. The buttons, and everything else that controller needs access to, are fields. They are fields so that methods in this class can be used by the Controller. For example, in the previous image before this one, the method `addNextListener(ActionListener actionListener)` is used to attach the appropriate ActionListener to the "next" button.


```

public class PatientModel {

    public ArrayList<Patient> patientList;

    public PatientModel()
    {
        patientList = new ArrayList<Patient>();
        Patient test = new Patient("01309123", "06/05/2000", "Johnson", "Hsiu");
        test.setTinAndCusDesc("he boutta die uh oh");
        test.setOccupation("noob");
        test.setEduDeg("elementary school");
        test.setWorkStatus("President");
        test.addMedicine(new Medicine("phosphorus", "johnson", "5 mg", "10 mg"));
        test.addMedicine(new Medicine("sodium", "johnson", "15 mg", "20 mg"));
        patientList.add(test);

        Patient test1 = new Patient("000001", "06/2/2000", "Marco", "Romagna");
        test1.setTinAndCusDesc("he die maybe in 2 days");
        test1.setOccupation("student");
        test1.setEduDeg("westmont high school");
        test1.setWorkStatus("good");
        patientList.add(test1);
    }
    public void addPatient(Patient patient)
    {
        patientList.add(patient);
    }
    public Patient getPatient(int index)
    {
        return patientList.get(index);
    }
}

```

This is a snippet for the PatientModel class in package Model. It simply keeps track of all the patients. There are some initialized test patients for easy testing.


```

*/
static class GotoInfoListener implements ActionListener {
    PatientDataPage db;
    boolean isCheckedSelected;
    Patient patient;

    public GotoInfoListener(PatientDataPage db) {
        this.db = db;
        isCheckedSelected = true;
    }
    public GotoInfoListener(Patient p)
    {
        patient = p;
    }

    public void actionPerformed(ActionEvent e) {
        if (isCheckedSelected == true) {
            if (db.isPatientSelected() == true) // then button will work
            {
                Patient patient = db.whichPatient();
                PatientInfoPage infoPage = new PatientInfoPage(frame, patient.getInfo());
                GotoDatabaseListener GotoData = new GotoDatabaseListener();
                GotoMedicationListener GotoMedication = new GotoMedicationListener(patient);
                infoPage.addBackListener(GotoData);
                infoPage.addNextListener(GotoMedication);

                infoPage.addUpdateListener((a) -> {
                    String[] info = infoPage.getEditedInfo();
                    patient.setPatientID(info[0]);
                    patient.setDateAdded(info[1]);
                    patient.setFirstName(info[2]);
                    patient.setLastName(info[3]);
                    patient.setBirthDay(info[4]);
                    patient.setGender(info[5]);
                    patient.setPhoneNum(info[6]);
                    patient.setSocialSecureNum(info[7]);
                    patient.setAddress(info[8]);
                    patient.setAddress2(info[9]);
                    patient.setCity(info[10]);
                });
            }
        }
    }
}

```

This is an interesting part of the code because it uses a lambda expression for the functional interface `ActionListener`. The lambda expression is used for the “Update” button when the user wants to update the information of a pre-existing patient. Everytime the update button is pressed, it will update the information in the model with the information in the view. This is used for every “Update” button.

ID	Date Added	FirstName	LastName	Birthday
01309123	06/05/2000	Johnson	Hsiung	03/02/1999
Address2	City	State	Zip	Country
NA	Santa Clara	CA	95051	USA

This screenshot displays patient information, and is done with a JTable which allows the user to edit the table in addition to viewing it. This fits perfectly for our needed functionality.

Select a Patient	
01309123	JohnsonHsiung
000001	MarcoRomagna ;)

Info

+ Add Visit

Visit History

Schedule Visit

Delete

This screenshot is for displaying the patients in the database, and is done with a JList which allows users to select an object. With this information, we can correctly match the info button, the +add visit button, the visit history button, the schedule visit button, and the delete button to the corresponding patient.

Testing Methods

```
//for testing purposes
// public static void main(String[] args) {
//     String s = "amazing nick";
//     VisitInfoPage n = new VisitInfoPage(new JFrame(), s);
// }
```

Some pages in the View package had their own main method. This main method just displays the one page we were working on. This was used for later pages where getting to them took some time.

As shown before. The PatientModel is initialized with some Patients for testing.

```
import static org.junit.jupiter.api.Assertions.*;

class PatientModelTester {

    @Test
    void testPatientModel() {
        PatientModel test = new PatientModel();
        assert test.getSize() == 2;
    }

    @Test
    void testAddPatient() {

        PatientModel test = new PatientModel();
        Patient patient = new Patient("01309123", "06/05/2000", "Johnson", "H
        test.addPatient(patient);
        assert test.getPatient(test.getSize()-1).equals(patient);
    }

    @Test
    void testGetPatient() {
        PatientModel test = new PatientModel();
        Patient patient = new Patient("01309123", "06/05/2000", "Johnson", "H
        test.addPatient(patient);
        assert test.getPatient(test.getSize()-1).equals(patient);
    }
}
```

This is the code snippet for JUnitTest for PatientModel. JUnitTests were only done for the model, because for classes in View, we cannot assert if the page looks correct. As for the controller, it did not really have any methods; it just had a lot of ActionListeners. JUnitTests were not done for getter and setter methods and simple methods because we thought they were unnecessary.

```
public void addPatient(Patient patient)
    patientList.add(patient);

public Patient getPatient(int index)
    return patientList.get(index);

public int getSize()
    return patientList.size();
```

Here are some examples of simple methods. They are just one line calling another method.

Deployment Instructions

Run the main method in PatientController.

Notes:

Required Input Registration: All fields in here are required. It will not allow you to goto the next page if some fields are not inputted.

Table: Information presented in tables can be edited directly. If after editing information, the cell is white, press enter so it turns blue. After, press "Update" in the center to update the information.

List: You have to select an object in the list to go to pages containing information of that category.

