AssignmentDFSW1504

Scrum

Name: B Karthik

MID:M1082972

Source Code:

https://github.com/BKarthik1/BKarthik1.git

Question

PROJECT STATEMENT

An organ bank management Application for an NOG where application can manage the recipient requests and process pipe line. In the application recipients can register themselves and Admin and Physician can track and approve the recipients.

- Recipient view
    - Recipient self-registration
- Admin view
    - Track Requests
    - View all Recipient details
- Physician View
    - List of Recipient registered
    - Approve Recipient after validating their medical profile

Functional Requirements

1. Recipient Registration & View
    a. Recipients can register themselves to the application.
    b. Recipient ID should be automatically generated
    c. They need to provide all their basic and medicinal profile.
2. Track Requests
    a. Admin can view all the recipient requests and close the requests.
    b. They need to see the status of all the Recipients along with other details – Approved / Not Approved.
    c. Admin can delete the Recipients whose profiles are not approved.
3. View Recipients
    a. List of all Recipient waiting in queue – able to filter based on demography and blood group.
4. Approve Recipients
    a. Physician can check all the recipients registered.
    b. They can moderate the recipient's profile and approve / reject.
    c. They can also raise queries if they need any more information to validate the profile.

UserStories Of Medical Stuff

Project Name: SMART **Company Name**: WELL HEALTH **Team Name**:

Changsheng Su (csu@ucsb.edu)

Jenny Aoyama (aoyama@ucsb.edu)

Huake He (huakehe@ucsb.edu)

Yuxiao Luo (yuxiaoluo@ucsb.edu)

Siyuan Wang (siyuanwang@ucsb.edu)

User Stories and use cases:

1) Log into the application

User Story: As a user, I can log into the application if I have a user account so that

I can interact with the virtual assistant and secure personal information**.**

Acceptance Criteria: User can enter the username and password onto the login   screen and submit for authentication. If the username and password did not pass the authentication, an error message will be displayed and the user can login again. If the username and password are correct, the users will be directed to the homepage.

Scenario 1: Authorized user log into the account.

- When the user starts the application, they will see a UI for them to log into their account.
- Once they are authorized, they can then access the app. **Scenario 2**: Unauthorized user log into the account
- When the user starts the application, they will see a UI for them to log into their account.
- If they are not authorized users, they will see a message showing they are unauthorized.

2)   Sign up for application

User Story: As a user, I can create my own account which stores my access   history and personal information.

Acceptance Criteria: For first time users, a sign-up option will also be available   to register as a new user. Users will insert their name, username, date of birth, email address, password, and other information to create an account. After the account creation is confirmed, the user will be directed to the login page. **Scenario 1:** New user can sign up for their account

- If the user doesn't have an account, they can click the sign up button, which will direct them to the sign up page.
- Then the user's account information will be stored in the database.

Scenario 2: New user signs up for the account that already exists.

- The sign up page will display a message showing the account already exists

3)   Change and Reset Password

User Story: As a user, I can change my password of my account if I believe my   personal information on the account can be at threat, or reset my password via email authentication if I forget my password.

Acceptance Criteria : Pass a manual test to see if change/reset password feature   can work via email authentication

Scenario 1: Users want to change password

- Users will see a button in account security page
- Users will answer three security questions, and use their email to authenticate their identity to change password

Scenario 2: Users fail to change password

- Users will see a button in account security page
- Users will answer three security questions, and use their email to authenticate their identity to change password
- Users fail to answer the security questions. Changing password fails.

4) Access previous appointment history and reports

User Story: As a user, I can go back into the history log to see the reports generated from past appointments, so that I recap the conversation with doctors after the meeting.

Acceptance Criteria: There is an appointment history and report history page for users to access, the timeline of past appointments are shown and reports generated that correspond to each appointment can be downloaded by clicking the 'detail' tab of the appointment.

Scenario 1: User check their previous appointment

- Users will see their previous appointments history in their dashboard.

Scenario 2: New user check their previous appointment

- Users will see their history empty if it is their first time scheduling the appointment

5) Check the patients' information

User Story: As a doctor, I can check the patients basic information.   Estimated time complete: 2 days.

Acceptance Criteria : A doctor is able to login their account and check the basic   information of the patients whom they have appointments with.

Scenario 1: Patients have their basic information.   ●

Doctors can access the patients dashboard.

● Doctors can send the request to get the patient's medical history.

Scenario 2: Patients don't have their basic information.   ●

Doctors can access the patients dashboard. ● Doctors will see

the empty history record.

6) Change the personal information

User Story: As a patient, I can change my personal information.

Acceptance Criteria   **:** Pass manual test of changing personal information in profile

Scenario 1: Users want to change their personal information in profile

- Patients will navigate to their profile page, and click change information button
- Patients will change the relevant personal information, and save to quit

7)   Close the session

User Story: As a patient, I can close the session to quit the transcription process.   **Acceptance Criteria   :** When patients complete their conversation with their doctors. They can quit the session, and the transcription task in the backend server will terminate.

Scenario 1: Users want to quit the session.

- Users can click the button to send a request to quit the transcription.
- Users will refer back to the homepage with the appointment reports updated. **Scenario 2:** No conversation is received.
- If no conversation is perceived in 15 secs, the server will terminate the transcription process.

8)   Extract the medical term

User Story: As a patient, I can request the server to extract the medical term from   a sentence.

Acceptance Criteria : When the user passes a sentence as an argument to the   server, the server should return extracted medical terms.

Scenario 1: Users expect to see extracted medical terms from a sentence which   contains medical terms.

- Users will see extracted medical terms returned from the server.

Scenario 2: Users expect not to see extracted medical terms from a sentence which   contains medical terms.

- Users will not see extracted medical terms returned from the server.

9)   Comprehend the medical term

User Story: As a patient, I can request the conversation to be analyzed by AI.   **Acceptance Criteria   :** When patients request the conversation to be analyzed, they   can send the request and the server will receive the scripts of the conversation, and analyze the scripts.

Estimated time to complete: 4 days

Scenario 1: Users want to analyze the conversation .

- Patients will send the request to analyze the conversation.
- Server will run the analysis algorithm to analyze the data and return the custom feedback.

Scenario 2: Users request for empty conversation script.

- Patients will send the request to analyze the conversation.
- Server will respond that the conversation is empty.

10) Highlight medical term

User Story: As a patient, I can request the server to highlight the medical terminologies.

Estimated time to complete: 3 days

Acceptance Criteria : When the server extracts the medical term, it can categorize it as a symptom or medicine, and highlight them for different colors..

Scenario 1: When medical symptoms are extracted

- Patients will see the medical symptom in the scripts highlighted to yellow.

Scenario 2: When medicine terminologies are extracted

- Patients will see the medicine terms being underscored.

Scenario 3: When nothing is extracted.

- Patients will see normal transcripts texts without any modification.

11) Getting the definition and explanation of the medical terminology

User Story: As a patient, I can send request to get the definition and explanation of the complex medical terminology

Acceptance Criteria : Pass manual test of inputting medical conversation audio and locating medical terms and send relevant information and explanation

Scenario 1: Users want to understand the medical term in conversation

- Patients will press a button to see the text transcribed from speech, and a list of medical terms contained.

- Patients will click on a specific term that he/she does not understand, and relevant information will be provided to help the understanding.

12) Explaining medical terms to the user

User Story: As a patient, I am able to see the pellucid explanation for medical   terms as my doctor tells me about my diagnosis result so that I won't have problems understanding these medical terms.

Acceptance Criteria : As users' conversations with doctors start, when a medical   term is mentioned by the doctor, it will be explained in real time to the user.

Medical terms are detected

- In real-time conversation, the AI will extract the medical terms and search for explanations for those terminologies.

Scenario 2: Medical terms are not detected

- The interface will then display the normal conversation scripts.

13) Access to patient's electronic health record

User Story: As a patient, I will be able to access my historical electronic health records so that the system will keep traces of the patient's historical health conditions.

Acceptance Criteria :   At the patient's end, the electronic health record will be   displayed in the user interface.

Scenario 1: Users have their current electronic health records   ● Users will

see a button for accessing the health record.

- Then they will see their EHRs from our database.

Scenario 2: New users don't have their electronic health records

- The AI will assist patients to fill out the important information for EHRs.

14) Ability to persist patient's electronic health record

User Story: As a patient my electronic health record should be persisted by the   application in the database.

Acceptance Criteria : The system will store each user's electronic health records, including reasons for appointment, diagnosis, doctor's advice, and prescriptions of medicine, in the database. The appointment history will also be stored into the database.

Scenario 1: Users want to store their electronic health records ● Patients

will see a button for storing EHRs.

● One window will pop up and will guide patients to store their EHRs.

Scenario 2: Users want to delete their electronic health records ●

Patients will see a button for removing EHRs.

● One window will pop up and ask users to confirm their choice.
● When users say yes, their EHRs will be removed.

15) Ability to update the patient's electronic health record

User Story: As a patient my electronic health record should be updated by the application and pushed to my EHR system via an API integration.

Acceptance Criteria : Any updates to a patient's EHR will be updated in the EHR used by the provider side via API integration with the EHR. **Scenario 1:** When patients finish their appointments

● AI will update the EHRs for patients. According to the conversation they have with doctors.

Scenario 2: Users want to update their electronic health records by themselves ● Patients will see a button to edit EHRs.

● One window will pop up and will guide patients to edit their EHRs.

16) Multilingual Interface

User Story: As a patient, if my native language is not English, I can switch the language of the content presented in the interface into the language that I am most comfortable with, which can be Spanish, Chinese, and etc.

Acceptance Criteria : This feature will be implemented by using an api like google translate which automatically generates translation. There will be a selection list button that contains all

supporting languages on the upper right corner of the webpage. The user can select a language and textual content on the webpage will then be translated into that language.

Scenario 1: Users are not comfortable with reading English ● Patients

will see a language option button.

- Then the user clicks it and will see a list of languages.
- Then the users will select their primary language.

17) Ability to generate a medical transcribed report post appointment User Story: As a patient, I will have a medical transcription report for the   appointment provided to me once my appointment has concluded.

Acceptance Criteria : This feature will generate a transcription report based on the   appointment voice recording using a speech to text converter and build out a medically transcribed report using NLP.

Scenario 1: Patients just finish their appointment with their doctors

- Patients will see a pop up window that concludes the conversation they had.
- It will also include the medical prescription and the follow up details for the future appointment.

18) Appointment Scheduling

User Story: As a patient, I can schedule an appointment with my doctors through a   virtual assistant.

Acceptance Criteria : When patients want to schedule an appointment, they don't   need to go through a long and misleading process. They can just tell our virtual assistant about their symptoms and our virtual assistant will handle the rest of the things for patients.

Scenario 1: Users want to schedule an appointment.

- Patients will see a button to launch the virtual assistant.
- Patients will see a virtual assistant will guide them to schedule the best appointment for them.

Scenario 2: Users fail to schedule an appointment.

- Patients will see a button to launch the virtual assistant.
- Patients will see an error message.

19) Remove patient from doctor's list

User Story: As a doctor, I can remove a single or multiple patients from my list.   **Acceptance**

Criteria **:** From the doctor's end, the user can select one or multiple   patients from his historical/upcoming appointment list and delete the patients. After the removal, the information of that patient will be permanently inaccessible from the doctor's end.

Scenario 1: Removal from upcoming appointment.

- Given that the patient is authorized
- The patient can cancel his/her appointment and the relevant information will be removed from the scheduled doctor's list.

Scenario 2: Removal from historical appointments

- Given that the doctor has authorized access
- The doctor can remove one or multiple patients in previous appointment list ● After the removal, the doctor will not be able to access that patient's data.

20) Display the appointment report

User Story: As a patient, I can send the request to retrieve the post appointment   report.

Acceptance Criteria **:** When patients want to check the post appointment record.

Scenario 1: Users want to receive the reports.

- Patients will see a button for getting the reports.
- After clicking the button, patients will see their profile is updated with new reports.

Scenario 2: No transcription task is required.

- Patients will see a button for getting the reports.
- The server will respond with a no transcription message to tell the users no reports are generated.

Technologies:

- Frontend: React: it gives us a user-friendly interface

- Backend: node-js: it allows us to organize all our microservices and allow the CORS policy,

- APIs: Amazon Medical Transcribe: it has high accuracy for medical field transcription, AWS Comprehend Medical: it has high accuracy extracting the medical terms.

- Database: Firebase: It reduce the work of implementing the database.