CS 460W – Software Development

Project Deliverable #2

Team Members: Alisa Vongsarasinh, Benjamin Bahadory, Daniel Baqaeen

3.1 **Functional Requirements**

3.1.1 **CreateRecord**

- 3.1.1.1 **Introduction** This function will mimic the responsibilities and the tasks done by an ER registrar. This function will prompt the user acting as the registrar through a list of required inputs. The inputs will be listed below. Only the registrar is allowed to access the CreateRecord function.
- 3.1.1.2 **Inputs** The inputs for this function will only be taken from the registrar user and they will be as follows: Name, Address, Number, Email, Data of Birth, Symptoms, Primary Physician, Health Insurance, Covid Vaccination Name (Pfizer, Moderna, or Johnson & Johnso) Covid Vaccination Date (1st shot and 2nd shot date but only one date for Johnson & Johnson. Also, date for booster), Emergency Contact Name, Emergency Contact Number, Allergies, Previous Medical Conditions, Previous Medications, Height, Weight, Religion, Race, Pregnancy, Gender, SSN, Sexually Active, Blood Type. and Check in Date.
- 3.1.1.3 **Processing** First a check will be done to make sure that the required values are non-empty. For each input the type will be checked as will, for example: for the number field, there will be a check to make sure only numbers are input in that field and not any other characters. This function will also create a unique patient ID.
- 3.1.1.4 **Outputs** Once the input values are verified, the user will be able to submit the inputs where they will be saved into a database. If there is an error in any of the inputs, an error message will appear indicating where the error is such as empty field or not correct input type.

3.1.2 **ViewRecord**

- 3.1.2.1 **Introduction** This function will be used just to view the data and not edit anything. The registrar will have limited access to only view general information, however, the nurse and the doctor will be able to access the medical information of the patient.
- 3.1.2.2 **Inputs** There are no inputs for this function since this is a view only function.
- 3.1.2.3 **Processing** This function will retrieve the correct record from the database by doing a query search for the patient's information like date of birth and name or by patient ID.

- 3.1.2.4 **Outputs** If there is no record with the provided information, then nothing is displayed. If there is a match then the information is displayed in an understandable and structured format.

3.1.3 **EditRecord**

- 3.1.3.1 **Introduction** This function will be used to edit the patients record in order to either update previous information or add data collected by the nurse or the doctor later on. This is accessible by the registrar, nurse, and doctor. Each will be allowed a different access and what information they can edit. For example, registrar can't view or edit medical information.
- 3.1.3.2 **Inputs** The inputs for this function will only be taken depending who the user is. If the user is the registrar then the inputs would be the same as for the CreateRecord function. If the user is the nurse, the inputs are as follows:

  - ➢ Vitals (blood pressure), Height, Weight, Nurse Notes, Admitted, Number of Nights, and Assigned Physician

  If the user is the doctor, the inputs are as follows:

  - ➢ Doctor Notes
  - ➢ Diagnosis:
    - ○ Covid-19 | Medication = Acetaminophen (lowers fever) ; Price = $6 for 24 500mg
    - ○ Asthma | Medication = Albuterol; Price = $15 for 1 90ml bottle
    - ○ Diabetes | Medication = Insulin; Price = $100 for 1 10ml vial
    - ○ Hypertension | Medication = Benazepril; Price = $20 for 30 10mg tablets
    - ○ Pneumonia | Medication = Clarithromycin; Price = $30 for 20 500mg tablets
    - ○ Heart Attack | Medication = Aspirin; Price = $10 for 30 81mg tablets
  - ➢ Requested Tests

    Laboratory Test:

    - ■ Hematologic Labs
      - ● Red blood cell
      - ● White blood cell
      - ● Liver function test
      - ● Renal function test
      - ● Electrolyte test
    - ■ Radiologic Labs
      - ● X-rays

  - CT scans
  - MRI scans
- Urinary Test
- Stool Test

Prescription:

- Injection
  - Intramuscular injection (IM)
  - Intravascular injection (IV)
  - Subcutaneous injection (SC)

- P.O (Per Os; Oral medication)

- 3.1.3.3 **Processing** The previous information for the patient is retrieved using a query, then the data is displayed and can be edited. Once the data is filled a check will be done to make sure that the required values are non-empty. For each input the type will be checked as will, for example: for the height input, there will be a check to make sure only numbers are input in that field and not any other characters.
- 3.1.3.4 **Outputs** Once the input values are verified, the user will be able to submit the inputs where they will be saved into the database. If there is an error in any of the inputs, an error message will appear indicating where the error is such as empty field or not correct input type.

3.1.4 **WriteDischargeInfo**

- 3.1.4.1 **Introduction** This function will be only accessible by the doctor and will contain information relating to the diagnosis, tests provided, medications, and other possible treatments if not medication.
- 3.1.4.2 **Inputs** The inputs for this function will be as follows:
  - Diagnosis, test provided, and medication
- 3.1.4.3 **Processing** Once the data is filled a check will be done to make sure that the required values are non-empty, for example Diagnosis.
- 3.1.4.4 **Outputs** Once the input values are verified, the user will be able to submit the inputs where they will be saved into the database. If there is an error in any of the inputs, an error message will appear indicating where the error is such as empty field. The discharge info will also be viewed in a detailed and structured format.

3.1.4 **ViewDischargeInfo**

- 3.1.4.1 **Introduction** This function will be only accessible by the patient. This will be where the patient can view the discharge info written by the doctor relating including the diagnosis, tests provided, and medication
- 3.1.4.2 **Inputs** There will be no inputs as this is just a viewing function
- 3.1.4.3 **Processing** This function will retrieve the correct information from the database by searching for the patient's ID and checking if the record where the discharge info is filled and not empty
- 3.1.4.4 **Outputs** If there is no record (discharge info yet) then a message is displayed stating that the discharge info is yet to be added. If there is a match then the discharge info is displayed in an understandable and structured format.

3.1.5 **CreateBill**

- 3.1.5.1 **Introduction** This function will be only accessible by the billing department. It will be responsible for going over the services provided and calculating what the patient owes.
- 3.1.5.2 **Inputs** The inputs will be the services provided like tests, medication, IVs and number of nights patient stayed if the patient stays overnight (date of check in and checkout) Also, the name, address, and patient details will be needed.
- 3.1.5.3 **Processing** There will be no check for empty values here since the patient may receive no services and will only be billed with the default visit charge. Also, a calculation will be done to find the total amount the patient owes. The prices for each service will be already stored and then depending on what services were provided, the patient will be charged accordingly, and a default visit charge will be always added. This data and the total amount due will be stored in the database.
- 3.1.5.4 **Outputs** Once the input values are verified, the user will be able to submit the inputs where they will be saved into the database. If there is an error in any of the inputs, an error message will appear indicating where the error is such as empty field. The bill will also be viewed in a detailed and structured format with the deadline and instructions on how to pay clearly stated.

3.2 **External Interface Requirements**

3.2.1 **User Interfaces**

- Login Screens: Every user will be presented with a hospital login screen, with a box for usernames and passwords, and a login button. Possibly a different portal for patients and employees, selected from a home page.
- Personal interfaces: Depending on credentials entered, users of different types (Patient, doctor, nurse) will be directed to an appropriate page containing their necessary functions.

    - Patient - Intake form: Input all necessary data for the hospital, including symptoms/chief complaint
        - Text boxes for input
        - Account page, shows bills
    - Nurse - Patient database: View all patient data and input new data such as vitals, notes, nights stayed, IV status. Doctors write prescriptions and give diagnoses, nurses help administer treatment and log it.
        - Lists for all information, text boxes for specific inputs. Buttons for confirmation and deletion.
        - Tabs with patient names: Click patient, opens customizable data table
    - Doctor - Patient database: View all patient data including info added by nurse. Add diagnosis and medication prescription.
        - Lists for all information, text boxes for specific inputs. Buttons for confirmation and deletion.
        - Nearly the same as nurse but with different input permissions
    - Billing - Input prices for each service, posts to patient account page

3.2.2 **Hardware Interfaces**

- Desktop and laptop computers (with their necessary peripherals) connected wirelessly or via ethernet to a common hospital network. Servers needed to host network and web pages.
- Screen-oriented terminal control

3.2.3 **Software Interfaces**

- Windows 10 or later, with Java Virtual Machine installed

3.2.4 **Communication Interfaces**

- Website hosted on servers, or an app which is only usable on computers on the hospital network, also hosted by servers.

3.3 **Performance Requirements**

- Static Requirements - One terminal per nurse and doctor, plus a few public terminals for patients who do not submit intake forms from their own machine.
- Dynamic Requirements - Logging in and submitting intakes and billing statements can take a few seconds or more, can take up to 10-15 seconds without causing any major disruptions. Editing patient data must be done quickly however, as it will need to be done by multiple accounts and at high frequency.

3.4 **Design constraints**

3.4.1 **Standards compliance** We will be following the HIPAA rules and regulations. We will be ensuring that the patient's data is protected and only viewed by the allowed parties. For example a patient's medical record will only be accessible by the doctor, nurse, and any other parties the patient gives permission to. As for report format, we will ensure that the bill will be in a detailed and understandable format for the patient.

3.4.2 **Hardware limitations** Enough memory should be dedicated to store all patients records (Petabytes of memory) and have a server that can handle all the requests and functions

3.5 **Attributes**

3.5.1 **Availability** The system should reasonably be available to any user 95% of the time, meaning that there is a 5% fault tolerance, or 18.25 days of the year where the system will be down. The availability relies heavily on the reliability of the software, hardware, and security.

3.5.2 **Security** As personal information will be inputted into the database, encryption can be implemented as well as activity logging to monitor for any suspicious activity, especially in areas where there needs to be a high level of security (e.g., billing area as credit card information will be needed).

3.5.3 **Maintainability**

- Traceability. The code will be able to be traced through its life, from its inception to development to deployment as well as through any changes made after deployment.
- Coupling. Based on the role of the user, certain items will be activated on the GUI (e.g., nurses and physicians will be able to see medical history, but the billing department will not) which can be altered if necessary. Since many aspects of the program are linked, those changes will be displayed in the update version of the medical report GUI.
- Cohesion. The classes used will only focus on their specific function making the program highly cohesive, especially due to the fact that it will be built with Java, an object-oriented language.
- Portability. Program will be available on any device that has internet access.

3.5.4 **Reliability** The system will be tested multiple times by all members of our team to ensure that our program works correctly.

Use Cases:

Use case name	Login
Participating actors	Initiated by Registrar, Nurse, Doctor, Billing staff, Patient to access the system
Flow of events	1. All of the users above activate the “login” function of a terminal that has access to the system 2. The user inputs the username, password, and the user type then clicks on the log in button 3. Connection is made to the database to check the credentials 4. If a match is not found then an error message will be viewed on the screen indicating that the credentials were not correct 5. If there is match then the user is redirected to the correct portal
Entry conditions	● Any of the users above has access to the system with the correct credentials
Exit conditions	● The user has the correct credentials to access the system, OR ● the user has the incorrect credentials and received the error message indicating that
Quality requirements	● Time out a user after 5 bad attempts within 24 hours

Use case name	Logout
Participating actors	Initiated by Registrar, Nurse, Doctor, Billing staff, Patient to disconnect from the system
Flow of events	1. The user clicks on the logout button 2. The connection is killed and the session is timed out
Entry conditions	● The user has to be already logged in
Exit conditions	● The logout button clicked and the session successfully timed out
Quality requirements	● This use case includes the login use case. The login use case is initiated when any of the users enter the correct credentials. The user must be logged in to access this use case

Use case name	Write Discharge Info
Participating actors	Initiated by Doctor
Flow of events	1. The Doctor activates the “Write Discharge Info” function by clicking on the write discharge info button from the portal 2. The Doctor fills the input boxes for Patient ID, Name, Date of Birth, Diagnosis: • Covid-19 • Asthma • Diabetes • Hypertension

	• Pneumonia • Heart Attack Tests provided: • Laboratory Test: o Hematologic Labs ▪ Red blood cell ▪ White blood cell ▪ Liver function test ▪ Renal function test ▪ Electrolyte test o Radiologic Labs ▪ X-rays ▪ CT scans ▪ MRI scans o Urinary Test o Stool Test • Prescription: o Injection ▪ Intramuscular injection (IM) ▪ Intravascular injection(IV) ▪ Subcutaneous injection (SC) o P.O (Per Os; Oral medication) Medications: • Acetaminophen • Albuterol • Insulin Benazepril • Clarithromycin • Aspirin 3. Once the Doctor fills in the information, the doctor clicks on the submit button
Entry conditions	● The user has to be a Doctor to view this function, AND ● Has to click on the write discharge info button
Exit conditions	● The Doctor clicks on the submit button
Quality requirements	● This use case includes the login use case. The login use case is initiated when a Doctor enters a correct credentials. The Doctor must be logged in to access this use case

Use case name	View Discharge Info
Participating actors	Initiated by Patient
Flow of events	1. The Patient activates the “View discharge info” function by clicking on the View discharge info button from the portal

	2. If the Doctor haven't activated the "Write Discharge Info" function for that patient then a message will be displayed saying discharge info is still not available 3. If the Doctor has already activated the "Write Discharge Info" function then the discharge info will be displayed 4. The patient can click on the download discharge info and a pdf will be downloaded containing the discharge information
Entry conditions	● The user has to be a Patient to view this function, AND ● Has to click on the view discharge info button
Exit conditions	● The Patient clicks on the download button, OR ● The Patient leave the View Discharge Info tab
Quality requirements	● This use case includes the login use case. The login use case is initiated when a Patient enters the correct credentials. The Patient must be logged in to access this use case

Use case name	Create Bill
Participating actors	Initiated by Billing Staff
Flow of events	1. The Billing Staff activates the "Create Bill" function by clicking on the Create Bill button from the portal 2. The Billing staff enters the Patient ID to retrieve the patient's information 3. The Billing department will have a view that includes all possible tests and medications (same as the ones mentioned in the doctor use case), they should also have an input for the number of nights patient stayed if the patient stayed overnight 4. Once the information above is filled in then the Billing Staff user presses the "continue" button 5. The next page will have input boxes for the patient's name and address 6. The Billing Staff then clicks submit button
Entry conditions	● The user has to be a Billing Staff to view this function, AND ● Has to click on the Create Bill button
Exit conditions	● Billing Staff clicks on the submit button
Quality requirements	● This use case includes the login use case. The login use case is initiated when a Billing Staff enters the correct credentials. The Billing Staff must be logged in to access this use case

Use case name	CreateRecord
Participating actors	Initiated by Registrar Communicates with Hospital Database

Flow of events	1. A patient and/or patient’s caregiver(s) walks in and towards the registrar area and requests to be seen by a doctor. 2. Registrar navigates to the hospital’s database GUI, hovers over the “Create Record” button, and clicks which displays an empty medical form. 3. Registrar prompts the patient or patient’s caregiver(s) to fill out the empty form with information such as: - Full Name - Date of birth - Address - Contact Information (Phone Number, Email) - Social Security - Gender, Race, Religion - Height, Weight, Blood Type - Primary Physician & Health Insurance - Emergency Contact Information (Name, Phone Number) - Medical History (Symptoms, Medical conditions & medications, Pregnancy, Covid Vaccination Type & Date) 4. Once form is filled, registrar checks that all fields were filled in correctly. 5. When finished with the information check, registrar will scroll to the bottom of the form and click the “Submit” button. 6. The newly created record will be uploaded to the hospital’s database and the hospital’s database GUI refreshes to display actions available to the registrar.
Entry conditions	● Registrar clicks the “Create Record” button on the hospital’s database GUI.
Exit conditions	● Registrar has checked that all fields are filled in correctly and clicks the “Submit” button which uploads the newly created medical record into the hospital’s database.
Quality requirements	● This use case includes the login use case. The registrar must log in with their authorized username and password to access the hospital database GUI that is specific for them.

Use case name	EditRecord (Registrar)
Participating actors	Initiated by Registrar Communicates with Hospital Database
Flow of events	1. Registrar accesses the hospital’s database on a web-based program on a hospital authorized device in the registrar area. 2. Registrar navigates to the hospital’s database GUI that displays the actions that are available to them. 3. Registrar clicks the “Edit Record” button which displays a search page and locates the patient’s record by inputting the

	patient’s first and last name into the search bar and by clicking the “Search” button. 4. When located, registrar will select the patient’s record and make any edits in the specific fields that they can edit. Specific fields include: - Full Name - Date of birth - Address - Contact Information (Phone Number, Email) - Social Security - Gender, Race, Religion - Primary Physician & Health Insurance - Emergency Contact Information (Name, Phone Number) 5. When edits are finished, registrar will scroll to the bottom of the record, hover over, and click the “Done” button. 6. A message will be displayed, cautioning that the record has been edited and asks whether or not the registrar would like to continue. 7. If registrar clicks “Yes” button, the page will refresh to display the newly updated medical record. If registrar clicks “No” button, the page will revert back to original medical record.
Entry conditions	● Registrar clicks the “Edit Record” button on the hospital’s database GUI.
Exit conditions	● Registrar clicks the “Yes” button when the caution message appears which refreshes the page to display the newly updated medical record. OR ● Registrar clicks the “No” button when caution message appears which reverts the edited record back to the original record.
Quality requirements	● This use case includes the login use case. The registrar must log in with their authorized username and password to access the hospital database GUI that is specific for them.

Use case name	EditRecord (Nurse)
Participating actors	Initiated by Nurse Communicates with Hospital Database
Flow of events	1. Nurse accesses the hospital’s database on a web-based program on a hospital authorized device in the patient’s room. 2. Nurse navigates to the hospital’s database GUI that displays the actions that are available to them. 3. Nurse clicks the “Edit Record” button which displays a search page and locates the patient’s record by inputting the

	patient’s first and last name into the search bar and by clicking the “Search” button. 4. When located, nurse will select the patient’s record and make any edits in the specific fields that they can edit. Specific fields include: - Vitals (e.g., blood pressure) - Height, Weight - Nurse Notes - IVs - Assigned Physician - Admitted Date and Number of days spent in hospital 5. When edits are finished, nurse will scroll to the bottom of the record, hover over, and click the “Done” button. 6. A message will be displayed, cautioning that the record has been edited and asks whether or not the nurse would like to continue. 7. If nurse clicks “Yes” button, the page will refresh to display the newly updated medical record. If nurse clicks “No” button, the page will revert back to original medical record.
Entry conditions	● Nurse clicks the “Edit Record” button on the hospital’s database GUI.
Exit conditions	● Nurse clicks the “Yes” button when the caution message appears which refreshes the page to display the newly updated medical record. OR ● Nurse clicks the “No” button when caution message appears which reverts the edited record back to the original record.
Quality requirements	● This use case includes the login use case. The nurse must log in with their authorized username and password to access the hospital database GUI that is specific for them.

Use case name	EditRecord (Doctor)
Participating actors	Initiated by Doctor Communicates with Hospital Database
Flow of events	1. Doctor accesses the hospital’s database on a web-based program on a hospital authorized device in the patient’s room. 2. Doctor navigates to the hospital’s database GUI that displays the actions that are available to them. 3. Doctor clicks the “Edit Record” button which displays a search page and locates the patient’s record by inputting the patient’s first and last name into the search bar and by clicking the “Search” button. 4. When located, doctor will select the patient’s record and make any edits in the specific fields that they can edit. Specific fields include:

	● Diagnosis: ○ Covid-19 ○ Asthma ○ Diabetes ○ Hypertension ○ Pneumonia ○ Heart Attack ● Tests Provided Laboratory Test: ○ Hematologic Labs ■ Red blood cell ■ White blood cell ■ Liver function test ■ Renal function test ■ Electrolyte test ○ Radiologic Labs ■ X-rays ■ CT scans ■ MRI scans ○ Urinary Test ○ Stool Test Prescription: ○ Injection ■ Intramuscular injection (IM) ■ Intravascular injection (IV) ■ Subcutaneous injection (SC) ○ P.O (Per Os; Oral medication) ● Doctor Notes 5. When edits are finished, doctor will scroll to the bottom of the record, hover over, and click the “Done” button. 6. A message will be displayed, cautioning that the record has been edited and asks whether or not the doctor would like to continue. 7. If doctor clicks “Yes” button, the page will refresh to display the newly updated medical record. If doctor clicks “No” button, the page will revert back to original medical record.
Entry conditions	● Doctor clicks the “Edit Record” button on the hospital database GUI.
Exit conditions	● Doctor clicks the “Yes” button when the caution message appears which refreshes the page to display the newly updated medical record. OR ● Doctor clicks the “No” button when caution message appears which reverts the edited record back to the original record.

Quality requirements	● This use case includes the login use case. The doctor must log in with their authorized username and password to access the hospital database GUI that is specific for them.

Use case name	RegistrarView
Participating actors	Initiated by Registrar Communicates with Hospital Database
Flow of events	1. Registrar accesses a computer with access to the hospital database and logs in with their credentials. 2. Registrar is directed to an account page with various options, including the option to view their patient information. 3. The patient clicks the link to their information page or list and all information except the doctor’s notes are available for them to review.
Entry conditions	● Registrar clicks the “My information” tab/link on their profile page
Exit conditions	● Registrar is linked to another page or terminates the program.
Quality requirements	● This use case includes the login use case. The registrar must log in with their authorized username and password to access the hospital database GUI that is specific for them.

Use case name	PractitionerView
Participating actors	Initiated by Doctor or Nurse Communicates with Hospital Database
Flow of events	1. Doctor/Nurse accesses hospital login page from a hospital-issued computer with access to the hospital network. 2. Doctor/Nurse navigates to the hospital’s database GUI that displays the actions that are available to them. 3. Doctor/Nurse selects the link or dropdown list of patient names and selects one. 4. When a patient is selected, their entire record will be available to view for the doctor or nurse.
Entry conditions	● Doctor or nurse clicks on a patient name or ID.
Exit conditions	● Doctor or nurse is redirected to a new page when another link is selected. ● The program is terminated.
Quality requirements	● This use case includes the login use case. The doctor or nurse must log in with their authorized username and password to access the hospital database GUI that is specific for them.

Use case name	BillingView
Participating actors	Initiated by Billing Department Employee Communicates with Hospital Database
Flow of events	1. An employee from the billing department accesses the hospital database by logging in to an authorized computer with their credentials. 2. Billing employee receives a list of patients who need to be billed for their services. 3. Selecting a patient reveals all medical services provided so that the billing agent can add the appropriate charges when creating a bill. 4. The billing agent can move forward with the billing process from links appearing on this page.
Entry conditions	● Billing agent accesses the patient database.
Exit conditions	● Billing agent clicks the link taking them to the create bill page. ● Billing agent terminates the program.
Quality requirements	● This use case includes the login use case. The billing agent must log in with their authorized username and password to access the hospital database GUI that is specific for them.

Use case name	ViewBill
Participating actors	Initiated by Registrar Communicates with Hospital Database
Flow of events	1. Registrar accesses the hospital's database on a web-based program on a hospital authorized device in the registrar area. 2. Registrar navigates to the hospital's database GUI that displays the actions that are available to them. 3. Registrar clicks the "View Bill" button which directs them to the page displaying their balance with instructions on how to pay.
Entry conditions	● Registrar clicks the "View Bill" button on the hospital's database GUI.
Exit conditions	● Registrar clicks the return button to return to their main profile page.

	● Registrar terminates the program.
Quality requirements	● This use case includes the login use case. The registrar must log in with their authorized username and password to access the hospital database GUI that is specific for them.