User Story Summaries:

The HIS is designed using multiple distinct user stories, which we explain each in this document:

Parmoun Khalkhali Sharifi:

User Story 1 (U.S - 7): Family doctor to get the patient's information and be searched

- Process Overview:
  - Nurses log in using a username and password, which redirects them to the nurse portal.
  - The portal features various options, including assigning a family doctor, managing consent forms, and viewing a list of patients.
  - To assign a family doctor, a nurse clicks on "Add Patient," which leads to a form that must be filled out with the patient's details.
  - At the bottom of the form, there's an option to either assign a family doctor and see this update in the database or skip this step, leaving the family doctor information empty.
- Testing and Verification:
  - The functionality has been tested with JUnit and end-to-end testing to ensure reliability.
  - The output demonstrates the successful assignment of a family doctor, including the ability to find patient details such as phone number, email, specialty, and name.
  - The system gracefully handles invalid inputs by displaying a "not found" message when a family doctor's credentials are not in the database.

User Story 2 (U.S - 8): Setting the patient's consent form status

- Process Overview:
  - After logging in, nurses access the nurse portal, where they can manage patient consent forms.
  - By clicking the designated button, a pop-up page prompts confirmation to consent on behalf of the patient.
  - Upon confirmation, the database is updated to reflect the consent, marked with a lowercase "t." If no consent is provided, it's indicated with an "f."
- Testing and Verification:
  - The feature's functionality has been validated with JUnit tests, ensuring the GUI framework correctly updates the database.

User Story 3 (U.S - 16): Prescribing Medication

- Process Overview:
  - Doctors log in with their credentials, gaining access to the patient portal.
  - Within the portal, clicking the "Prescribe Medication" button opens a pop-up for entering patient ID and the prescribing doctor's name.
  - Additional medication details, such as dosage and usage instructions, are entered.
  - Submitting the form confirms the prescription, and the user can return to the dashboard by closing the prescription portal.
- Testing and Verification:
  - The medication prescribing process has been thoroughly tested end-to-end.
  - The updated table and GitHub indicate that JUnit tests confirm the system works correctly and updates the database via the GUI.

Mehregan Mesgari

User Story 1: Add lab request on the physicians page

- Process Overview: The physician can request a lab on their page for any patient using their patient ID. After log in, the physician will click on request lab, add patient ID, add the lab type and put a description for the lab
- Testing and verification: This method is tested end to end. The peer review and test cases written verify this functionality works correctly.

User Story 2:Retrieve lab results on the physicians page

- Process Overview: After the lab fulfills the lab test, the physician can access the lab result on their home page. After the doctor logs in, they can click on review lab results, search the patient by their ID and retrieve all the lab results
- Testing and verification: This method is tested end to end. The peer review and test cases written verify this functionality works correctly.

User Story 3: Email family doctor with patient information after discharge

- Process Overview: After the patient is discharged, if they had a valid email for their family doctor and gave consent, their family doctor will receive an email with full visit information in their mailbox
- Testing and verification: This method is tested end to end. The peer review and test cases written verify this functionality works correctly.

User story 4: Laboratory dashboard

- Process Overview: The laboratory dashboard has 2 components. Checking new lab requests and fulfilling labs. Firstly the lab tech can click on the new lab requests and view all the labs that are added since they last logged in. Their second option is to fulfill labs. They can do so by clicking on fulfill lab results, search the patient by their ID, fulfill the lab by clicking on the lab and clicking on add lab results

- Testing and verification: This method is tested end to end. The peer review and test cases written verify this functionality works correctly.
-

Gaurav Charan Moturi:

User Story 1 (U.S - 1): Hire Physician so there are checks

- **Process Overview:**

1) Select Admin role

- **User selects the “Admin Login” option**

2) Admin logs in with correct credentials

3) Admin selects option to either “Hire Physician” or “Hire Nurse”

4) Admin fills in all required fields, including setting up a username and password.

- **Admin requires first name, last name, age, specialty, gender, address, username and password to create a new Physician/Nurse to add to the hospital system.**

5) Admin can now hire Physician/Nurse and their information will be saved in the database when they press the “Add Physician/Add Nurse" button.

End-to-End Testing: The functionalities of hiring a physician or nurse has been tested comprehensively, covering general cases and corner cases.

Cost Estimate: 12 hours.

Hire Physician refactored by Gaurav

Corner cases:

- ➢ **No first name:** Tested without inputting first name.
- ➢ **No last name:** Tested without inputting last name.
- ➢ **No age:** Tested without inputting age.
- ➢ **No gender:** Tested without inputting a gender
- ➢ **No username:** Tested without inputting a username
- ➢ **No password:** Tested without inputting a password
- ➢ **First or last name with numbers added:** Tested by adding numbers to first or last name.
- ➢ **Age is not an Integer value:** Tested without inputting an integer value for age.
- ➢ **Gender must be “Male” or Female”:** Tested by inputting any other string into the gender section.

Acceptance Criteria:

Ensure that the first name and last name are both strings of letters. Ensure that the age is an integer value. Ensure that the gender is either Male or Female. Ensure that there is a username and password.

User Story 2 (U.S - 9): Requesting Lab on the Physician's Page

- **Process Overview:**

1) **Physician Login and Dashboard Access:**
    a) **The physician logs into the system using their unique credentials.**
    b) **Upon successful login, the physician is directed to their dashboard, where they have multiple options, including patient management, prescriptions, and lab requests.**
2) **Navigating to Lab Request Module:**
    a) **From the physician's dashboard, the option to "Request Lab" is selected, bringing up the 'AddLabRequestGUI'.**
    b) **This interface is designed for physicians to submit lab requests for their patients easily.**
3) **Filling Out Lab Request Form:**
    a) **The physician fills in the required fields to request a lab test. This includes selecting a patient from their list, choosing the type of lab test needed, and any specific instructions or notes relevant to the lab test.**
    b) **The form is designed to ensure all necessary details are captured to process the lab request accurately.**
4) **Submission and Database Update:**
    a) **Once the form is completed, the physician submits the lab request.**
    b) **The system validates the input to ensure completeness and correctness.**
    c) **Upon successful validation, the lab request is added to the database, making it available for the laboratory staff to access and fulfill.**
5) **Notification and Follow-Up:**
    a) **The physician receives confirmation that the lab request has been successfully submitted.**
    b) **The laboratory staff processes the request, and the results are uploaded to the system.**
    c) **The physician is notified when the lab results are available, enabling them to review the results and take appropriate action.**

End-to-End Testing: The lab request functionality has been tested, including form validation, database integration, and notification processes.

Cost Estimate: 15 hours.

Corner cases:

- ➢ **Missing patient selection:** The system prompts the physician to select a patient.
- ➢ **Incomplete test details:** The system requires all fields to be filled before submission.
- ➢ **Invalid instructions:** The system validates the instructions for prohibited characters.

Acceptance Criteria:

- ➢ Ensure that physicians can easily navigate to and use the lab request module.
- ➢ Ensure all necessary information for a lab request is captured accurately.
- ➢ Ensure the system validates input to prevent errors in lab requests.
- ➢ Ensure lab requests are correctly added to the database for laboratory staff to access.

User Story 3 (U.S - 15): Registration, Login, and Login Validation Module

- **Process Overview:**

2) **User Selection and Access:**
    a) **User launches the application and is greeted with a Role Selection interface.**
    b) **User chooses their role (Admin, Lab, Physician, Nurse) and selects the "Login" option accordingly.**
3) **Registration Process:**
    a) **Admins must register new users (Physicians or Nurses) through the “Hire ‘Role’” button.**
    b) **This opens a form and requires essential information such as first name, last name, age, gender, address, username, and password. For Physicians, an additional field called specialization is required.**
    c) **Upon submission, the system validates the input for completeness and correctness (all fields entered properly as required).**
    d) **Successful registration adds the user to the system database, and they are prompted as such and directed to the same page again for admins to enter more information if required.**
4) **Login Process:**
    a) **Existing users select the "Login" option and are prompted to enter their username and password.**
    b) **The system uses the LoginValidator.java to authenticate the credentials against the database records.**
    c) **Successful login grants access to role-specific functionalities and dashboards (Physician's Dashboard, Nurses's Dashboard, Admin Dashboard, Lab Dashboard).**
5) **Validation Checks:**
    a) **The system checks for incorrect inputs, such as wrong username and password.**

End-to-End Testing: Comprehensive testing includes successful registration and login, error handling for incorrect credentials, and role-specific access restrictions.

Cost Estimate: 18 hours.

Corner cases:

- ➢ **Incorrect username and password:** Validation ensures only correct usernames and passwords are accepted past the login page.
- ➢ **Unregistered users attempting to log in:** The system prompts Login Failed and will not allow the user past the login page.
- ➢ **Unauthorized role access:** Users are restricted to features and areas specific to their role and unable to access other features.

Acceptance Criteria:

- ➢ Ensure user registration captures and validates all required fields.
- ➢ Ensure user login correctly authenticates and directs users based on their role.
- ➢ Ensure user experience is seamless between registration, login, and accessing role-specific features.

Amira Mohamed:

User Story 1 (U.S - 6): Finding a patient and displaying their information.

- **Process Overview:**

1) **Login as Doctor:**
    - The physician accesses the system by logging in using the "RoleSelectionGUI" with their credentials.
2) **Patient Portal:**
    - Upon successful login, the physician is directed to the patient portal page.
3) **Display all patients:**
    - In the patient portal, the physician can view all the patients assigned to them by selecting the “Display all patients” option.
4) **Get patients by their ID:**
    - In the patient portal, the physician has the option to retrieve patient information by entering their ID. This is done by selecting "Get patient by ID" and entering the patient’s ID, after which the information will get displayed on the screen.
5) **Get patients by their name:**
    - Similarly, the physician can retrieve patient information by entering their name. This is done by selecting "Get patient by name" and providing the patient’s name, which will then display the patient's information.

- **Testing and Verification:**

End-to-end testing: The functionality of displaying patient information undergoes comprehensive testing to verify its performance under general and edge cases.

- **Display all patients:** Displays only the patients assigned to the physician, ensuring confidentiality and relevance.
- **Get patient by ID:** Always returns one patient due to the uniqueness of patient IDs. Patients not assigned to the physician will not be displayed even if a valid ID is entered.
- **Get patient by name:** May return multiple patients if their first or last names match the entered query.

This testing ensures that the physician can access and search for specific patient information assigned to them.

Integration Testing:

- Integration testing verifies the seamless interaction between the patient portal GUI and the database operations.
- It ensures that the GUI accurately communicates with the database when searching and retrieving patients' information.

Cost Estimation: 30 hours of development + 10 hours for testing and refactoring
Actual Time Spent: 35 hours of development + 10 hours for testing and refactoring

User Story 2 (U.S - 5): Record and retrieve the vital signs of the patient

- **Process Overview:**

1) **Login as Doctor:**
    - The physician logs in using the "RoleSelectionGUI" with their credentials.
2) **Patient Portal:**
    - After logging in, the physician is directed to the patient portal page.
3) **Record Vital Signs:**
    - In the patient portal, the physician enters the patient's ID, along with vital sign information such as temperature, systolic pressure, diastolic pressure, heart rate, and respiratory rate.
    - After entering the information, the physician clicks on the "Record Vital Signs" button.
    - The system stores the vital sign information in the database or updates it if the patient's records already exist.
4) **Retrieve Vital Signs:**
    - The physician can also retrieve vital signs for specific patients based on their ID.
    - Upon entering the patient's ID and clicking on the appropriate button, a panel displaying the patient's vital signs is shown.

- **Testing and Verification**

Constructive Junit testing and end-to-end testing to test vital signs class.

JUnit Testing:

- JUnit tests were conducted for the class responsible for vital signs to ensure their correctness and robustness.
- Test cases were designed to cover various scenarios, including valid and invalid inputs, boundary cases, and exceptional conditions.
- Assertions were used to verify the expected behaviour of methods under different circumstances.

End-to-End Testing: The functionalities of recording and retrieving vital signs are tested comprehensively, covering general cases and corner cases.

Corner cases:

- ➢ **Invalid Patient ID:** Test with a negative number as the patient ID.
- ➢ **Non-Existing Patient ID:** Verify behaviour when entering a patient ID that does not exist in the database.
- ➢ **Incorrect Data Type for Temperature:** Test with incorrect data type input for temperature.
- ➢ **Invalid Temperature Range:** Check if the system handles input of temperatures outside the acceptable range (30-45°C).
- ➢ **Invalid Systolic and Diastolic Pressure Range:** Test with values outside the acceptable range (0-200 mmHg).
- ➢ **Invalid Heart Rate Range:** Verify behaviour for heart rate values outside the acceptable range (0-200 bpm).
- ➢ **Invalid Respiratory Rate Range:** Test with respiratory rate values outside the acceptable range (10-60 breaths/min).

Acceptance Criteria:

Ensure that only existing patient IDs can store and retrieve vital signs.
Validate that temperature, systolic pressure, diastolic pressure, heart rate, and respiratory rate fall within their respective acceptable ranges.

Cost Estimation: 25 hours of development + 10 hours for testing and refactoring
Actual Time Spent: 30 hours of development + 10 hours for testing and refactoring

User Story 3 (U.S - 12): Discharge patients from the hospital

- **Process Overview:**
  1) **Login as Doctor:**
     - The physician logs in using the "RoleSelectionGUI" with their credentials.
  2) **Patient Portal:**

    - After logging in, the physician is directed to the patient portal page.

    3) **Discharge patients:**
    - In the patient portal, the physician can choose a patient from the drop-down menu and click on "Discharge selected patient" to discharge the patient from the hospital.
    - If there are no patients in the system, the discharge functionality will be disabled

- **Testing and Verification**

End-to-End Testing: The functionality of discharging patients is tested comprehensively to check the database and GUI, covering general cases and edge cases:

- The patient GUI first checks if there are any patients in the database. If so, the discharge patient functionality will be enabled, and the physician will be directed to the "Discharge patient" page. If not, it displays an informative message indicating that there are no patients to discharge.
- In the "DischargePatient GUI", when the physician selects a patient, they are removed from the drop-down menu, and the drop-down menu gets updated with the patients that are in the database.
- The code checks if there are still patients in the system when updating the drop-down menu and displays that there are no remaining patients if there are no more patients in the database.

Integration Testing:

- Test the integration between the patient portal GUI and the database operations.
- Ensure that the GUI interacts correctly with the database when discharging patients.
- Verify that changes made in the GUI are reflected in the database, such as removing the discharged patient.

Edge Cases and Boundary Testing:

- Discharge the last patient in the database and verify that appropriate messages are displayed.
- Discharge patients when the database contains only one or a few patients.

Cost Estimation: 10 hours of development + 5 hours for testing and refactoring
Actual Time Spent: 15 hours of development + 10 hours for testing and refactoring

Harrish Elango:

User Story 1 (U.S - 1/2): Hire Physician/Nurse

- **Process Overview:**

1) **Select Admin role**
    - ○ **User selects the “Admin Login” option**
2) **Admin logs in with correct credentials**
3) **Admin selects option to either “Hire Physician” or “Hire Nurse”**
4) **Admin fills in all required fields, including setting up a username and password.**
    - **Admin requires first name, last name, age, specialty, gender, address, username and password to create a new Physician/Nurse to add to the hospital system.**
5) **Admin can now hire Physician/Nurse and their information will be saved in the database when they press the “Add Physician/Add Nurse" button.**

End-to-End Testing: The functionalities of hiring a physician or nurse has been tested comprehensively, covering general cases and corner cases.

Hire Physician refactored by Gaurav

Corner cases:

- ➢ **No first name:** Tested without inputting first name.
- ➢ **No last name:** Tested without inputting last name.
- ➢ **No age:** Tested without inputting age.
- ➢ **No gender:** Tested without inputting a gender
- ➢ **No username:** Tested without inputting a username
- ➢ **No password:** Tested without inputting a password
- ➢ **First or last name with numbers added:** Tested by adding numbers to first or last name.
- ➢ **Age is not an Integer value:** Tested without inputting an integer value for age.
- ➢ **Gender must be “Male” or Female”:** Tested by inputting any other string into the gender section.

Acceptance Criteria:

Ensure that the first name and last name are both strings of letters. Ensure that the age is an integer value. Ensure that the gender is either Male or Female. Ensure that there is a username and password.

User Story 2 (U.S - 3/4): Resign Physician/Nurse

- **Process Overview:**

1) **Select Admin role**
    a) **User selects the “Admin Login” option**
2) **Admin logs in with correct credentials**
3) **Admin selects option to either “Resign Physician” or “Resign Nurse”**
4) **Admin can go through a list of existing Physicians and Nurses in the hospital system, and choose which one they want to resign.**

5) Selected Physician/Nurse will be resigned from Hospital system when admin presses "Remove Selected Physician/Nurse" button

End-to-End Testing: The functionalities of resigning a physician or nurse have been tested comprehensively, covering general cases and corner cases.

Acceptance Criteria: As long as there is a physician or nurse to resign, this GUI is operable. The list of Physicians/Nurses will be empty if there are no Physicians/Nurses currently hired by the hospital.

User Story 3 (U.S - 17): Assign Physician and current Nurse to Patient in NurseAddGUI

- **Process Overview:**

1) **Select Nurse role**
   - **User selects the "Nurse Login" option**
2) **Nurse logs in with correct credentials**
3) **Nurse selects "Add patient" button in NurseGUI**
4) **Nurses must fill in all relevant fields for adding a patient into the system. This is also where the nurse can choose from all of the options for a physician to assign to the patient being admitted.**
5) **Nurse presses "Add Patient" button which adds the new patient to the hospital system, as well as to the Physician chosen and the nurse admitting the patient.**

End-to-End Testing: The functionalities of assigning a physician to the patient and the current nurse to the patient has been tested comprehensively.

Acceptance Criteria: Assigning a physician and the nurse to the patient requires that the patient is added correctly. This requires that the nurse fill each text field in the add patient frame and choose one of the physicians from a list of Physicians hired by the hospital to assign to the patient. Admitting the patient into the hospital also assigns that patient to the Nurse that fills out the patient's information.

User Story 4 (U.S - 18): Display general information about Patients being taken care of by Nurse

- **Process Overview:**

1) **Select Nurse role**
   - **User selects the "Nurse Login" option**
2) **Nurse logs in with correct credentials**
3) **Nurse selects "View Patients" button in NurseGUI**
4) **Nurses are now able to see which patients in the hospital system have been assigned to them.**

End-to-End Testing: The functionality of seeing all patients assigned to the current nurse has been tested comprehensively to make sure no other nurse's patients appear, and only patients assigned to the current nurse appear.

Cost Estimate: 9 hours.

Refractored Developer Stories:

1) **Fixed Physician/Nurse ID issue when populating list for resigning. ID number now properly matches Physician or Nurse Employee ID.**
2) **Fixed Resign Physician/Nurse function so that it deletes based on EmployeeID. No longer deletes multiple physicians if they have the same names.**
3) **Fixed nurse assignment to patient when adding patient to hospital system.**
4) **Refactored the assignment of physicians to patients from random to selected.**
5) **Made patient gathering dependent on the Physician's ID as well as the Nurse's ID so we know which patients are assigned under each employee.**
6) **Added "Sign Out" buttons on each main GUI page to return to RoleSelectionGUI.**

Link to other documents:

Bug configuration and class refactoring:
https://docs.google.com/document/d/1vuzcxLJhiPnCYpfaGo8mG83GN-1Cvggm6v6y3LxoD64/edit

Planning Doc:
https://docs.google.com/document/d/1oyA1L4V6MDJ_Ccv_4-wr7w6yoowYAcSZf6i3pw12wFA/edit