Navsangeet Kaur

CSCI 370

Software Engineering

Dr. Svadlenka

Valid Test Plan and Result

Introduction

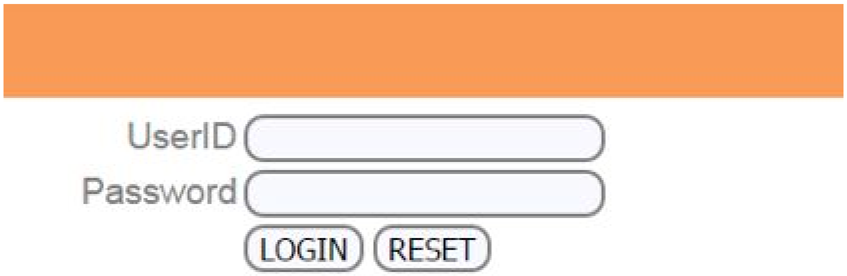
Development Testing is necessary to discover bugs/faults before the software delivery to the client/users. It guarantees the quality of the software as well as ensures ease of use, high level performance, and reliable software operations. The quality of the product which requires lower maintenance cost and results in providing accurate, consistent, and reliable results. Thus, it is an essential part of the Baby Heart system to produce accurate heartbeat so there’s no confusion that can create panic. For instance, if there is an error that outputs that the heartbeat is not detected, it can cause stress to the mother/user which can be harmful for the baby. On the other hand, if the system outputs a positive result which is regular heartbeat if the baby but in reality the heartbeat is abnormal or not detected, it can extend the time for the mother to know about the abnormalities for immediate consultation with the doctor. Therefore, to make the software bug free and reliable, I use three types of development testing that are Unit, Component, and System.

Unit

Unit testing examines individual units or object classes. It also supports testing the functionality of the classes or objects. The Baby Heart system lets the user have access to their result and data using the login system. The user needs to login the system with the correct username and password. The system uses hashMap to match the attempted input by the user with the password in the hash table that was made during registration. When the user input the correct login information they redirect to the main page of the app/system. The expected and actual outputs of this test are “The main webpage of the system,” and “login successful.”

Moreover, the unit testing also helps in proper error handling when the inavlid input is filled in by the user. For instance, when the user doesn’t enter a username and/or password, the expected and actual output that the system displays is “Please enter username and password to log in.”

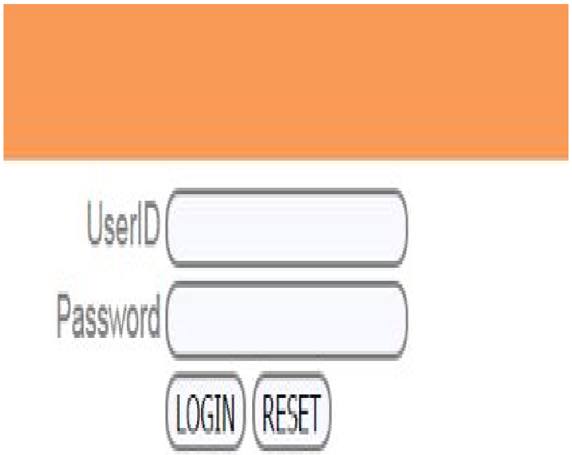
On the other hand, in the Baby heart system, the user fills in an invalid username or password the system displays the message, “The username or password you’ve entered are incorrect, create an account” and gives them another attempt to have access to the system. If the inputs are still correct then the expected and actual output that the system displays is “ Invalid username and password, forgotten password?,” the system gives the user an attempt to input their username and password again or give them an option to click “forgotten password?” to let them reset their password.

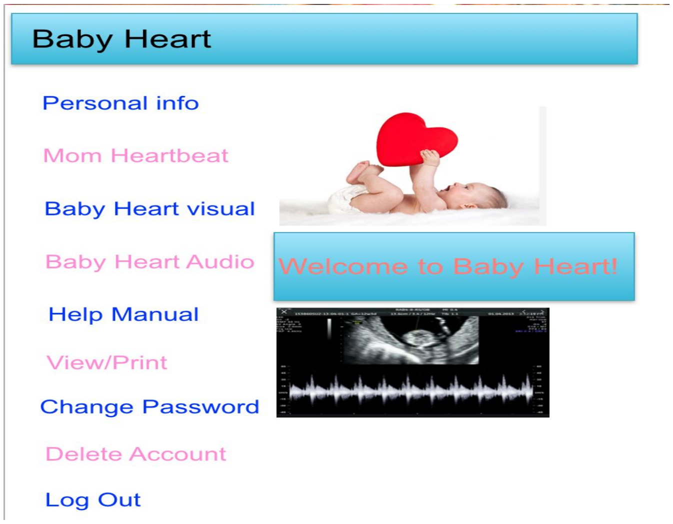


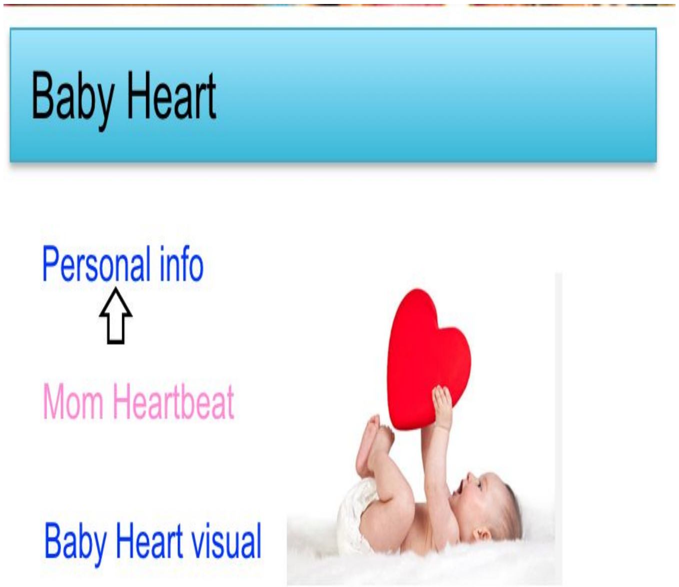
Component

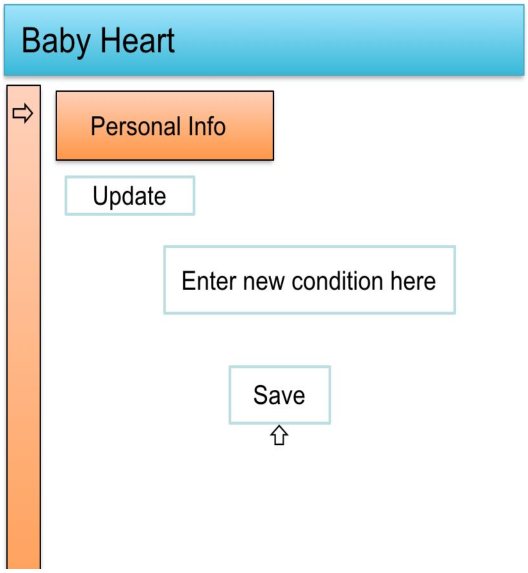
The component testing supports verifications of the interactions using message-passing interfaces among the various subsystems of architecture. It should test the correct functioning of the interaction under normal operating conditions. The interactions in the system are: Interfaces between Login and menu module, personal information and the edit, update, and delete menus, Mom heartbeat and view/print menu, Baby heart visual, audio and view/print menu, print menu to delete menu/trash module, view to download menu, download to print menu, and change password to the login update information. The component testing also used by the subsystem to interact through messages such as the optical light sensors send a message to the system that the heartbeat is detected, it receives that message and asks the device if arrhythmia is detected and the system sends back the message if it is detected or not. Similarly, the LED light sensor detects the heartbeat from 90-110 bpm or 140-170 bpm. Also, the transducer detects the heartbeat of the fetus. These two devices interact with each other and match the beats per minute and then send the message to the system if the heartbeat is detected or not or if the heartbeat is arrhythmic, abnormal heartbeat per minute.

For instance, in the Baby Heart system, the component testing checks the interface link between the Login and the menu module. The user enters the login credentials and clicks on the login button. The expected result is the user directs to the menu page of the system. Then the user clicks on personal information and changes her health condition information in that module, the system successfully opens up the new blank column where the user enters new condition/allergies and clicks on the save button. Then, the updated information should appear in the saved personal information.


UserID
Password
LOGIN
RESET


Baby Heart
Personal info
Mom Heartbeat
Baby Heart visual
Baby Heart Audio
Help Manual
View/Print
Change Password
Delete Account
Log Out
Welcome to Baby Heart!


Baby Heart
Personal info
Mom Heartbeat
Baby Heart visual


Baby Heart
Personal Info
Update
Enter new condition here
Save

Furthermore, the component testing handles the timing errors for the interaction when invalid input is received by the system. In the Baby Heart system, using the above testing area, the component testing makes sure when the user logs in the system using login credentials and is successful then the main page shows the menu page on the left and in the center it displays "Welcome to Baby Heart!" In addition, it also handles the error when the user logs in the system, and the main page is failed, it gives the user a reload option to click and reload the link and display the correct interfacing page.

System

System testing validates the fully integrated software product. It also tests all the priority functional requirements to evaluate the end-to-end system specification. It verifies the functionality through testing of all inputs to check for the expected output. This testing is performed on the whole developed system and software product before it is introduced to the clients and/or end users. The system specifications requirements are divided into two parts: functional and non-functional requirements. The functional requirements are numbered through 1-8 and non-functional requirements are organized with alphabetical letters for better traceability. Each functional requirement below is separated into 1, 2, or 3 parts to show each case in detail with positive and negative scenarios, error handling inputs, and decision-making approach. The system testing for some of the functional requirements in the business specification are as below:

3. Case #1: The user enters the username and password and gets into the system, then the user sees edit permissions to modify the personal information and save it then

log out of the system. The expected result is: The user is logged into the home page and the edit screen is presented to the user when clicked on the edit option in the personal information and the edited information is saved and then upon logging out, the system exits the user out of the account.

8. Case #1: Positive test: The system outputs a message, “Heartbeat of the Baby,” when a second heartbeat is detected by the LED device, contrasting to the user’s heartbeat.

Case #2: Negative test: The system outputs a message, “No Baby Heartbeat,” when the second heartbeat isn’t detected for 3 seconds.

Case #3: Negative test: The system also detects if the heartbeat is abnormal using the regular heartbeat per minute chart for the pregnant women and the fetus. It matches the information with the detected heartbeat and displays the message to the system, “Arrhythmia is detected.”

f. Case #1: The username field must take a minimum of 8 characters, a maximum of 10 characters, numbers (0-9), letters (a-z, A-Z), special characters (such as exclamation, hyphens, pound sign, etc.) and it can’t be left blank.

Case #2: The password field must take a minimum of 6 characters, a maximum of 8 characters, numbers(0-9), letters (a-z, A-Z), special characters (such as exclamation, hyphens, pound sign, etc.) and it can’t be left blank.

Case #4: If the wrong credentials are entered, it should indicate that to the user and reload the login page.

Case #5: If the user enters the correct credentials, it should take the user to the next user interface.

Case #6: If the user enters the correct credentials but wishes to cancel the login, then it should not take the user to the next UI and reload the login page.