



MUHAMMAD FAZEEL ASHRAF
01-235161-077

Automated Biological Lab

Bachelor of Science in Information Technology

Supervisor: Usman Shafique

Department of Computer Science
Bahria University, Islamabad

December 2019

Certificate

We accept the work contained in the report titled “Automated Biological Lab”, written by Muhammad Fazeel Ashraf as a confirmation to the required standard for the partial fulfillment of the degree of Bachelor of Science in Information Technology.

Approved by . . . :

Supervisor: Usman Shafique

Internal Examiner: Dr. Faisal Bashir

External Examiner: Dr. Majid Iqbal Khan

Project Coordinator: Dr. Muneeb Gohar

Head of the Department: Dr. Muhammad Muzammal

December 16th, 2019

Abstract

Our present era is going through a major technological shift. Everything from minor to major is shifting towards technology to ensure ease of use through software applications and automation systems. In the current age of fast-growing technology, system followed in our medical domain (i.e. medical labs) follow the old and tiresome methods for preparing diagnostic reports and associated medical lab facilities. The goal of this project is to facilitate lab technicians by automating lab report management and patient record tracking systems, saving time and effort, compared to now as the current system requires manual entries of the reports and patient records.

Acknowledgments

Firstly, I would like to praise the Almighty ALLAH (SWT) and thank him who has given me the strength and knowledge to complete this project to the best of my ability.

Secondly, I would like to pay my humble regards to my mother and my brothers. Without their constant moral and financial support it wouldn't have been possible to achieve what I have today.

Lastly, I'm grateful to my supervisor Mr.Usman Shafique for his guidance and supervision throughout this project. Thank you, sir, for all the hard work, time, as well as your approach to my shortcomings and support you've invested with me in this project.

May Allah bless all of you.

MUHAMMAD FAZEEL ASHRAF
Islamabad, Pakistan

December 2019

*“A genius is simply one who has taken,
full possession of his own mind and directed it toward objectives,
of his own choosing without permitting outside influences,
to discourage or mislead him.”*

Napoleon Hill

Contents

1	Introduction	1
1.1	Project Background	1
1.2	Problem Description	2
1.3	Project Objective	2
1.4	Motivation and Challenges	2
1.5	Project Scope	3
1.5.1	Registration	3
1.5.2	Login System	3
1.5.3	Menu Operations	3
1.5.4	Report Management	3
1.5.5	Online Payment System	4
1.5.6	Finance Management	4
1.5.7	Record Management	4
1.5.8	View Report	4
1.5.9	Reviews and Ratings	4
2	Literature Review	5
2.1	Literature Review	5
2.1.1	Limit LIS	6
2.1.2	ELAB	6
3	Requirement Specifications	7
3.1	Existing System	7
3.2	Proposed System	7
3.3	Functional Requirements	8
3.4	Non-Functional Requirements	8
3.4.1	Availability	8
3.4.2	Usability	8
3.4.3	Maintainability	9
3.4.4	Response Time	9
3.4.5	Robustness	9
3.4.6	Security	9
3.5	Use Case Diagram	9
3.6	Descriptive Use Cases	10
3.6.1	Registration of Admin	11
3.6.2	User Login	12
3.6.3	Manage Patient Record	13

3.6.4	Calculate Bill	14
3.6.5	Provide Feedback	15
3.6.6	Pay Bill	16
3.6.7	Generate Report	17
4	Design	18
4.1	Context Flow Diagram	19
4.2	Activity Diagram	20
4.3	Entity Relationship Diagram	21
4.4	Sequence Diagram	22
4.5	Graphical User Interface	23
4.5.1	Main Screen	23
4.5.2	Add Test	24
4.5.3	Reports List	25
4.5.4	Add Invoice	26
4.5.5	List of Patients	27
4.5.6	Payment Page of 2 CheckOut	28
4.5.7	Reviews And Rating	29
5	System Implementation	30
5.1	Tools and Technologies	30
5.1.1	HTML	30
5.1.2	CSS	30
5.1.3	Bootstrap	30
5.1.4	JavaScript	31
5.1.5	JQuery	31
5.1.6	Visual Studio Code	31
5.1.7	Adobe Illustrator	31
5.1.8	GIT	31
5.1.9	MEAN Stack	31
5.1.10	MongoDB	32
5.1.11	Express	32
5.1.12	Angular	32
5.1.13	NodeJS	32
5.2	Implementation Strategy	32
5.3	Functionalities of the System	33
6	System Testing and Evaluation	34
6.1	Graphical User Interface Testing	34
6.2	Usability Testing	34
6.3	Software Performance Testing	34
6.3.1	Volume Testing	34
6.3.2	Stress Testing	35
6.4	Compatibility Testing	35
6.5	Exception Handling	35
6.6	Load Testing	35
6.7	Security Testing	35

6.8	Installation Testing	35
6.9	Test Cases	36
6.9.1	Login Test Case	36
6.9.2	Add New Patient Test Case	36
6.9.3	Sign-Up A New Admin Test Case	37
6.9.4	Select Medical Tests Test Case	38
6.9.5	Calculate Bill	39
6.9.6	Bill Payment Test Case	39
6.9.7	Feedback Test Case	40
6.9.8	Generate Report Test Case	41
6.9.9	View Report Test Case	42
7	Conclusions	43
7.1	Future Work	43
	References	45

List of Figures

3.1	Usecase Diagram	10
4.1	Context Flow Diagram	19
4.2	Activity Diagram	20
4.3	Entity Relationship Diagram	21
4.4	Sequence Diagram	22
4.5	Main Screen	23
4.6	Add Test	24
4.7	Report Lists	25
4.8	Add Invoice	26
4.9	List of Patients	27
4.10	Online Payment Page	28
4.11	Contact Us	29

List of Tables

2.1	This table is showing the comparison between the existing systems. . . .	5
2.2	This table is showing that ABL have more features from ELAB and Limit LIS.	6
3.1	Use Case: Registration of Admin	11
3.2	Use Case: User Login	12
3.3	Use Case: Manage Patient Record	13
3.4	Use Case: Calculate Bill	14
3.5	Use Case: Provide Feedback	15
3.6	Use Case: Pay Bill	16
3.7	Use Case: Generate Report	17
6.1	Test Case: Login	36
6.2	Test Case: Add New Patient	37
6.3	Test Case: Sign-Up A New Admin	37
6.4	Test Case: Select Medical Tests	38
6.5	Test Case: Calculate Bill	39
6.6	Test Case: Bill Payment	40
6.7	Test Case: Feedback	40
6.8	Test Case: Generate Report	41
6.9	Test Case: View Report	42

Acronyms and Abbreviations

ABL	Automated Biological Lab
API	Application Programming Interface
HTML	Hypertext Markup Language
CSS	Cascading Style Sheet
MEAN	MongoDB, Express, Angular, Node
DB	Database
JS	JavaScript
JSON	JavaScript Object Notation
GUI	Graphical User Interface
UI	User Interface

Chapter 1

Introduction

The automated biological lab is a web-based system, implemented for the automation of medical lab facilities including automated generation of medical reports, patients test reports, history of records, online payment, and commission management system.

1.1 Project Background

The procedure followed today in small labs is the patients are prescribed medical tests manually written by the doctor. Lab in charge takes appropriate blood samples and payment from the patients and manually creates a patient profile including his personal details, contact details and further attributes e.g. test name, doctor who referred the patient for the required test, etc. The patient profile is either saved in a register or on an excel sheet in both cases it's done manually. Pathologists perform tests on the blood samples and the results are noted down in the excel sheet. The lab in charge then generate reports by manually entering test details like reference values and patients profile. As life is becoming chaotic with time it's hard for the patients to come and collect reports especially if the hospital/medical lab is at a far distance. Secondly, as some of the medical labs offer services individually (i.e. are not owned by a hospital or its trademark) so there is some kind of agreement between individual practicing doctors and these labs. On every patient referred, a certain percentage of commission is given to the doctor by the lab. This procedure is maintained manually at both ends, the doctors and the labs. To address these issues of inconsistency, human error and inefficiency our system provides a way to resolve all these problems.

1.2 Problem Description

People nowadays are more aware of their surroundings and things that are going on around us. Besides, they have become more health-conscious. With the increasing success and discoveries in medical sciences and with modern times doctors have taken a modern approach to diagnose and cure disease. They have become more concerned with finding the core issue rather than just giving medicines to fulfill their duty of saving lives. To ensure that the disease is diagnosed in its early stages, they follow the routine practice of prescribing tests according to the diagnosis. Several pathology labs exist for this sole purpose. However, these small labs lack efficiency and there can be a human error that can be fatal in this domain and wastage of resources all because the system is handled manually. ABL facilitates our registered doctors, pathologists, and lab technicians to effortlessly maintain a lab report management system, patient record tracking on both doctor dashboard as well as in laboratory database and the patient can pay online and the commission will be calculated for the doctors.

1.3 Project Objective

Primary objectives of developing this system are:

- To design and develop a web application to automate the manual tasks performed in medical labs.
- To provide patients and their respective doctors with online report access.
- To provide ease for the small labs to manage their data.
- To maintain the medical diagnostic history of the patients.

1.4 Motivation and Challenges

Labs that are linked with hospital usually have modern technology and tools to carry out their tasks so our goal is to target the small medical labs that are working separately so that the system of medical labs becomes almost equal for all types of operating labs. The biggest challenge this project will face is the lack of technical knowledge among the users in Pakistan and marketing of this project is difficult, as majority users to some extent are reluctant to adopt the new system and shift from their traditional ways to the modern and latest systems and they are not familiar with those, because they have not seen such technologies that are used nowadays. Moreover building the trust of customers will be a challenging task because in our society it is still unusual in these modern times for a common man to pay online as there are certain myths about the misuse of credit cards and

frauds. For the users to trust the system and removing these myths is highly important. So I integrated the 2Checkout system which is globally used by many companies.

1.5 Project Scope

Major features of the system are:

1.5.1 Registration

This feature will allow the user to register in the system by signing up. The system will then prompt the user to enter personal, contact and login in details. E.g. name, age, gender, email, number, lab name and password.

1.5.2 Login System

The system will provide separate login portals for admins (lab technicians), Patients and doctors to log in their respective modules with their username and password. This feature will allow the users to login with their credentials and access the system accordingly.

1.5.3 Menu Operations

The menu operations will allow the user to view the menu which will include the option of viewing the basic system functionalities. This operation will also allow the users to log in into their respective domain i.e. admins (lab technicians), patients and doctors. Each user will be provided with a unique dashboard to perform respective functionalities according to their type.

1.5.4 Report Management

This feature is being developed mainly for lab technicians. It will include the automated generation and management of lab reports. This module will hold a list of all the medical tests including their reference values, cost, the prerequisite details needed for the test according to the age and gender of the patient. Admin will select the test from a dropdown and some details will be attached to it. e.g.while some tests are performed empty stomach. Upon filing the details and choosing the test, the system will automatically gather data and generate an invoice. The technician will only have to fill the obtained reference values. Moreover, an email notification will be sent to the patient upon report completion.

1.5.5 Online Payment System

This feature of the system provides online payment to improve flexibility for the user in case the user doesn't hold the exact amount of cash needed for the bill payment. Users are eligible to pay through credit cards to view their reports.

1.5.6 Finance Management

This feature deals with the financial aspects of the system. Based on the selection of medical tests, the system will keep adding the amount of the medical test along with the medical test name in the bill upon pressing the generate bill button.

1.5.7 Record Management

This feature is concerned with tracking the patient's medical history for example kind of tests the patient has been prescribed previously and by which doctor etc. The database will be allowed access to concerned personnel only for updating and modification.

1.5.8 View Report

This feature allows the users to view the test reports generated by the admins online from anywhere, the user only has an internet connection. This feature will be embedded in the mobile application. The users will only be able to view their reports.

1.5.9 Reviews and Ratings

This feature is added to provide value to the customers and to know customer's experience and feedback about the services provided to the labs which are using our system.

Chapter 2

Literature Review

It is important to understand and explore knowledge about existing systems to provide a better solution. This is helpful to differentiate between the already existing system and the system which I will be making. It is initiated to get a deeper knowledge of the problems which helps us in providing a good solution and compare my system quality to the existing system.

2.1 Literature Review

There are many existing systems based on new and emerging technologies that are related to this system. We have researched these existing systems but none of them implements all the features in the one same system. Some systems only provide report management, some lack maintains patient's profiles and so on. Their GUI's are old fashioned. The following table shows the analysis of our system with other current implemented systems.

Table 2.1: This table is showing the comparison between the existing systems.

Features	Existing Systems	
	Limit LIS	ELAB
Maintain Dependent Profile	No	Yes
Doctor's Dashboard	Yes	No
Online Report Viewing & Downloading	Yes	Yes
Finance Management	No	Yes
Online Payment	No	No
Automatic Report Generation	Yes	Yes
Reviews & Rating	No	No

2.1.1 Limit LIS

LimitLIS is a cloud-based medical laboratory solution that enables physicians and laboratory staff to organize their workflows and manage operations. Modules and key features include data entry screens, patient portals, audit trails, process automation, and reporting. LimitLIS's patient portal enables patients to view and download reports from the website. The portal also allows laboratory representatives to send diagnostics reports to patients via SMS and email. Physicians to validate reports with the help of digital signatures. But it lacks an online payment system, reviews, and rating feature and doctor's dashboard [1].

2.1.2 ELAB

ELAB, provided by Tru-Solutions, is a medical diagnostic center and lab management software. It provides features and functionalities related to our systems such as patient registration, referral management, patient information management, procedure-based billing, and a lab-instrument interface. eLab provides user-based access controls to provide authorized access for other features and admin rights. But it doesn't provide online report viewing and downloading, online payment system and reviews and rating features [2]. On the other hand, if I compare this system to the other related systems in the market, my system is covering all the features listed in the table which other systems are lacking.

Table 2.2: This table is showing that ABL have more features from ELAB and Limit LIS.

Automated Biological Lab	
Features	Automated Biological Lab
Maintain Dependent Profile	Yes
Doctor's Dashboard	Yes
Online Report Viewing & Downloading	Yes
Finance Management	Yes
Online Payment	Yes
Automatic Report Generation	Yes
Reviews & Rating	Yes

Chapter 3

Requirement Specifications

3.1 Existing System

The procedure followed today in small labs is the patients are prescribed medical tests manually written by the doctor. Lab in charge takes appropriate blood samples and payment from the patients and manually creates a patient profile including his details, contact details and further attributes e.g. test name, doctor who referred the patient for the required test, etc. The patient profile is either saved in a register or on an excel sheet in both cases it's done manually. Pathologists perform tests on the blood samples and the results are noted down in the excel sheet. The lab in charge then generate reports by manually entering test details like reference values and patients profile. As life is becoming chaotic with time it's hard for the patients to come and collect reports especially if the hospital/medical lab is at a far distance. Secondly, as some of the medical labs offer services individually (i.e. are not owned by a hospital or its trademark) so there is some kind of agreement between individual practicing doctors and these labs. On every patient referred, a certain percentage of commission is given to the doctor by the lab.

3.2 Proposed System

By automating this system, as the patient comes with the list of tests that doctors have prescribed, the sample will be taken and a receipt will be generated with a unique receipt no. and that will be given to the patient and will be saved in the system's record as well. Moreover, the patient has a choice whether he/she wants to pay the charges by cash or if they don't have enough cash they can also pay by card and they don't have to come later for taking the reports. It can be seen on the website. The patient can see their reports from anywhere. When the reports are ready an email will be generated to the patient that your

reports are ready if he/she didn't submit the money they can't see the reports until they do. The doctor's commission will also be generated according to the total charges of the report for a specific patient. In addition to this, the doctor can also see his patient reports by logging into the system.

3.3 Functional Requirements

- The system shall direct users to respected login portal and dashboard depending upon the user type like admins (lab technicians), patients and doctors.
- The system shall automatically display all the parameters of the selected medical test.
- System shall be able to calculate bills by adding charges of all the selected medical tests in the database.
- System shall be able to calculate the commission of the doctor depending upon the prefixed (assigned at the time of adding a doctor) percentage of the total amount of the bill.
- The system shall sent a notification by email or contact number to the patient when reports are ready.
- Patients shall be able to view his medical reports by signing in to the system.
- Upon logging in, the system shall allow the doctor to view his commission amount (income) and reports of his recommended patients only.
- The user shall be able to provide his feedback about his experience with the lab.

3.4 Non-Functional Requirements

3.4.1 Availability

For availing the system's services, the user must have an active internet connection.

3.4.2 Usability

The system shall be user-friendly. On average, 80 percent of users shall be able to interact with the system successfully in the first attempt.

3.4.3 Maintainability

The system shall be maintainable by the developer (super admin), to fulfill the requirements of the users, in case of an error or bugs are reported.

3.4.4 Response Time

The system's response to the user's query will be in maximum 10 seconds if server and internet connection both are in running state.

3.4.5 Robustness

The system shall display dialogue boxes to guide the user if an unexpected user input is triggered.

3.4.6 Security

The system shall prevent any unauthorized access to the system data and the system functions. No other than the users with access privileges shall be able to add, modify or delete the patient's history.

3.5 Use Case Diagram

Following is the use case diagram of the system. It has four actors named as admin, patient, doctor and the system. All the system-user interactions which are either performed by the user or the system have been mentioned. There are some processes that require further functionalities to be performed for their completion or have sub-processes associated with them are linked using extend and the include functions.

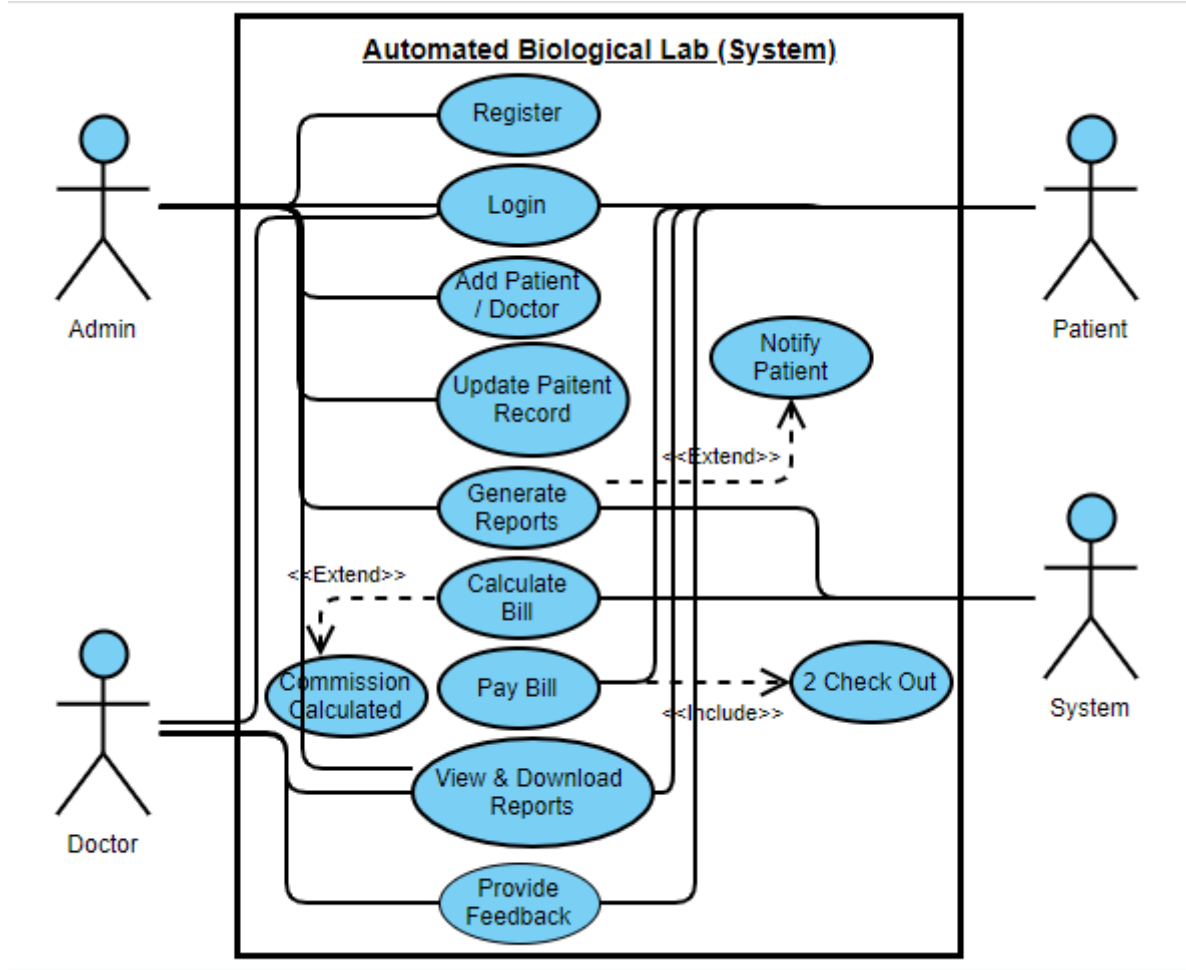


Figure 3.1: Usecase Diagram

3.6 Descriptive Use Cases

Use cases describe the flow of interaction-response between the system and the user. The following are the detailed use cases of the system written in tabular form, in which the actors, preconditions, postconditions and the normal flow of the process/ task are stated. Moreover, the alternative flow of the use case, exceptions and other special requirements (if any) are also mentioned in the table of use cases.

3.6.1 Registration of Admin

The descriptive explanation of the registration of admin process is shown in table 3.1.

Table 3.1: Use Case: Registration of Admin

Use Case ID:	UC-01
Use Case Name:	Registration of Admin.
Actors:	Admin.
Description:	Admin will enter the required data into the sign-up form. The system will authenticate that all the required data is provided or not. After validation by the system, the data of the admin will be saved into the system and the admin's account has been created.
Trigger:	"Sign Up" button.
Preconditions:	<ol style="list-style-type: none"> 1. The admin must have access to the internet. 2. The admin must have opened the sign-up window on the system.
Postconditions	The admin has been registered.
Normal Flow:	<ol style="list-style-type: none"> 1. The admin clicks the "Sign Up" button. 2. Sign up window is opened by the system. 3. Admin enters his data in the sign-up form and creates a password for his login account. 4. The admin clicks the "sign-up" button. 5. Data of the admin is stored in the database of the system. 6. The system displays a message "Congratulations! Your account has been created successfully".
Alternative Flows:	<p>In step 3 of the normal flow, if the patient left any mandatory field blank, then:</p> <ol style="list-style-type: none"> 1. The system will show an error. 2. A message is displayed "All the mandatory fields should be filled". 3. Admin fills all the required fields and process flow is resumed from step 4.
Expectations	None
Special Requirements / Other Information:	None

3.6.2 User Login

The descriptive explanation of the user login process is shown in this table 3.2.

Table 3.2: Use Case: User Login

Use Case ID:	UC-02
Use Case Name:	User Login
Actors:	Admins(Lab Technician)
Description:	The user shall enter his login id and password and choose the type of his/her. After authentication, the system will allow the user access to the respective admin module of the system.
Trigger:	Tapping the login portal button on the main page of the system.
Preconditions:	The admin is already verified and his credentials are stored in the system.
Postconditions:	The user has been successfully logged in to the system as an admin.
Normal Flow:	<ol style="list-style-type: none"> 1. The user initiates the use case by clicking on the portal button on the main screen. 2. The system displays the login portal of the user. 3. The user selects the portal according to his category. 4. Systems display a login window. 5. The user enters his/her credentials (user id, password). 6. The system authenticates the user id and password provided by the user. 7. After authentication, the system allows access to the user and directs the user to the respective dashboard.
Alternative Flows:	<p>In step 3 of the normal flow if the user has entered wrong admin id or password then:</p> <ol style="list-style-type: none"> 1. The system will generate an error. 2. The system will again ask the user to re-enter his/her admin id or password. 3. The user will enter his id and password again. 4. The use case will continue from step 3 of the normal flow if authenticated.
Expectations:	None
Special Requirements:	None

3.6.3 Manage Patient Record

The descriptive explanation of the manage patient record process is shown in the table 3.3.

Table 3.3: Use Case: Manage Patient Record

Use Case ID:	UC-03
Use Case Name:	Manage Patient Record
Actors:	Admin.
Description:	This use case let the admin keep a record of the patients to maintain patients' profile and his medical history and to avoid creating a record of the same patient every time a patient visits the laboratory.
Trigger:	Clicking the option of "Add New Patient" on the menu.
Preconditions:	All the necessary details have been provided by the patient.
Postconditions	<ol style="list-style-type: none"> 1. The patient's record has been saved in the system's database. 2. A unique ID number is assigned to the patient and the password is also generated.
Normal Flow:	<ol style="list-style-type: none"> 1. The user selects the "Add New Patient" option from the menu. 2. The user enters all the required details of the patient and clicks on the save button. 3. The system validates if all the mandatory fields are filled or not. 4. If all the mandatory fields all filled, then the system saves the patients' records in the database and generates a slip with a unique ID number and an auto-generated password. 5. The user clicks on the "OK" button and the use case is terminated.
Alternative Flows:	<p>In step 3 of the normal flow, if the patient left any mandatory field blank, then:</p> <ol style="list-style-type: none"> 1. The system will show an error. 2. A message is displayed "All the mandatory fields should be filled". 3. Admin fills all the required fields and process flow is resumed from step 4 of the normal flow.
Expectations	None
Special Requirements / Other Information:	The system shall assign a unique ID number to each patient and password for the first time.

3.6.4 Calculate Bill

The descriptive explanation of the calculate bill process is shown in the table 3.4.

Table 3.4: Use Case: Calculate Bill

Use Case ID:	UC-04
Use Case Name:	Calculate Bill
Actors:	System
Description:	The system will calculate the bill for the patient.
Trigger:	Tapping the button of the “Generate Bill” button on the menu.
Preconditions:	The lab technician has selected all the tests requested by the patient.
Postconditions	The lab technician has selected all the tests requested by the patient.
Normal Flow:	<ol style="list-style-type: none"> 1. The user will initiate the use case by tapping on the “Generate Bill” button on the menu. 2. The system will fetch the amount added in the inventory for each selected test and will calculate and display the total bill of the patient.
Alternative Flows:	None
Expectations	None
Special Requirements / Other Information:	The system shall calculate the commission of the doctor according to the pre-fixed percentage of the total amount of bill.

3.6.5 Provide Feedback

The descriptive explanation of the reviews and rating process is shown in the table 3.5.

Table 3.5: Use Case: Provide Feedback

Use Case ID:	UC-05
Use Case Name:	Provide feedback.
Actors:	Patient , Doctor
Description:	The patient and doctor can provide feedback about his experience about the labs which are using our system.
Trigger:	Clicking on the “feedback” option from the menu.
Preconditions:	<ol style="list-style-type: none"> 1. The user has logged in to the system. 2. The user has at least used the system one time.
Postconditions	The user has been successfully provided his feedback.
Normal Flow:	<ol style="list-style-type: none"> 1. The user selects the “Reviews & Ratings” option from the menu. 2. The system displays 5 stars for rating and an optional text box for compliments. 3. The user clicks on the number of stars according to his experience and adds compliments if he wants. 4. The number of stars selected by the user turns green. 5. The user clicks on the “OK” button and his feedback is submitted.
Alternative Flows:	None
Expectations	None
Special Requirements / Other Information:	None

3.6.6 Pay Bill

The descriptive explanation of the pay bill process is shown in the table 3.6.

Table 3.6: Use Case: Pay Bill

Use Case ID:	UC-06
Use Case Name:	Pay Bill
Actors:	Patient
Description:	The patient will pay the bill through 2 Checkout payment method by using the credit card.
Trigger:	Pay bill button.
Preconditions:	<ol style="list-style-type: none"> 1. The patient must have access to the internet. 2. The patient must have logged in to the system. 3. The user must have a credit card and amount in it for payment.
Postconditions	<ol style="list-style-type: none"> 1. Bill has been paid. 2. The patient can view his reports online and can download the reports as well.
Normal Flow:	<ol style="list-style-type: none"> 1. The user clicks on the “Pay Bill” button. 2. The user is redirected to 2 checkout payment page. 3. The user enters the data for paying the bill online. 4. The user clicks the “continue” button. 5. The system validates the user input. 6. Transaction of the calculated bill amount is made from the user account and receipt of the payment is generated. 7. The patient is redirected to the view reports page.
Alternative Flows:	<p>In step 5 of the normal flow, if the user left any mandatory field blank, or enters wrong input in any field, then:</p> <ol style="list-style-type: none"> 1. The system will show an error message. 2. The user fills all the required fields and provides the correct details and process flow is resumed from step 6.
Expectations	None
Special Requirements / Other Information:	None

3.6.7 Generate Report

The descriptive explanation of the generate report process is shown in the table 3.7.

Table 3.7: Use Case: Generate Report

Use Case ID:	UC-07
Use Case Name:	Generate Report
Actors:	Admin
Description:	The admin selects an invoice from the list of pending invoices and prepares the medical report of the patient against the invoice according to the obtained results of the test.
Trigger:	Selecting an invoice from the list of invoices where status is pending.
Preconditions:	<ol style="list-style-type: none"> 1. The user has performed the tests on the samples. 2. The user has an active internet connection and is logged in to the system.
Postconditions:	<ol style="list-style-type: none"> 1. The user has been successfully prepared the medical report of the patient. 2. The selected invoice status is changed to pending to complete from the list of invoices.
Normal Flow:	<ol style="list-style-type: none"> 1. The admin selects an invoice from the list of pending invoices. 2. The selected invoice is opened in a new window. 3. Click on the “Generate Report” button from the opened invoice window. 4. The report template page is opened by the system containing the reference values for the selected medical tests according to the invoice (a new page for each test). 5. The user enters the obtained values in the field of actual values. 6. The user clicks the “OK” button. 7. The reports are saved, and the patient is notified via email.
Alternative Flows:	None.
Expectations	None
Special Requirements / Other Information:	None

Chapter 4

Design

This section describes the architecture and the design of the system. It specifies the modules of the system, interaction between them and the user interface of the system to meet the requirements.

4.1 Context Flow Diagram

This diagram shows the overall structure of the ABL system. Admin, patient and the doctor are the participants (external entities) that are interacting with the system. Information exchanges between the participants and the system is shown in the figure 4.1

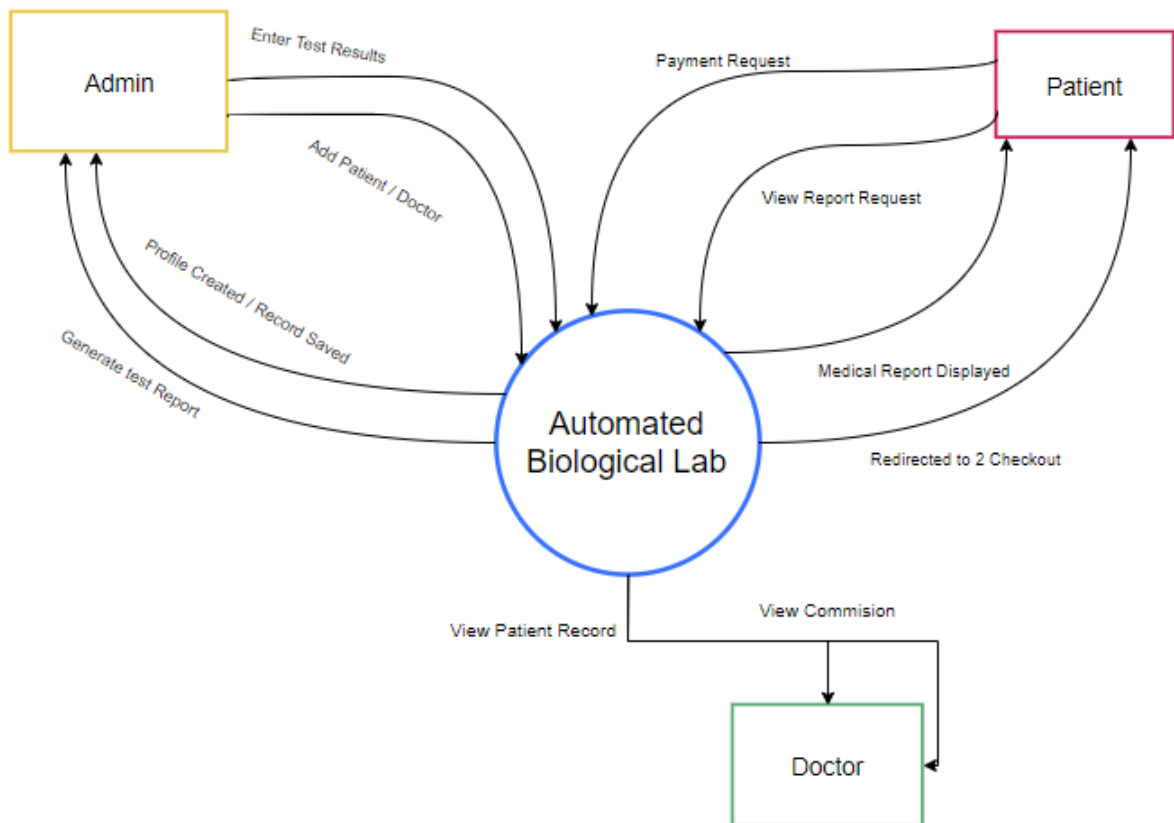


Figure 4.1: Context Flow Diagram

4.2 Activity Diagram

Following is the activity diagram of the Automated Biological Lab System in which the overall flow of the system is shown in the figure 4.2. This diagram illustrates the activities and the flow of behavior of the system when these activities are performed.

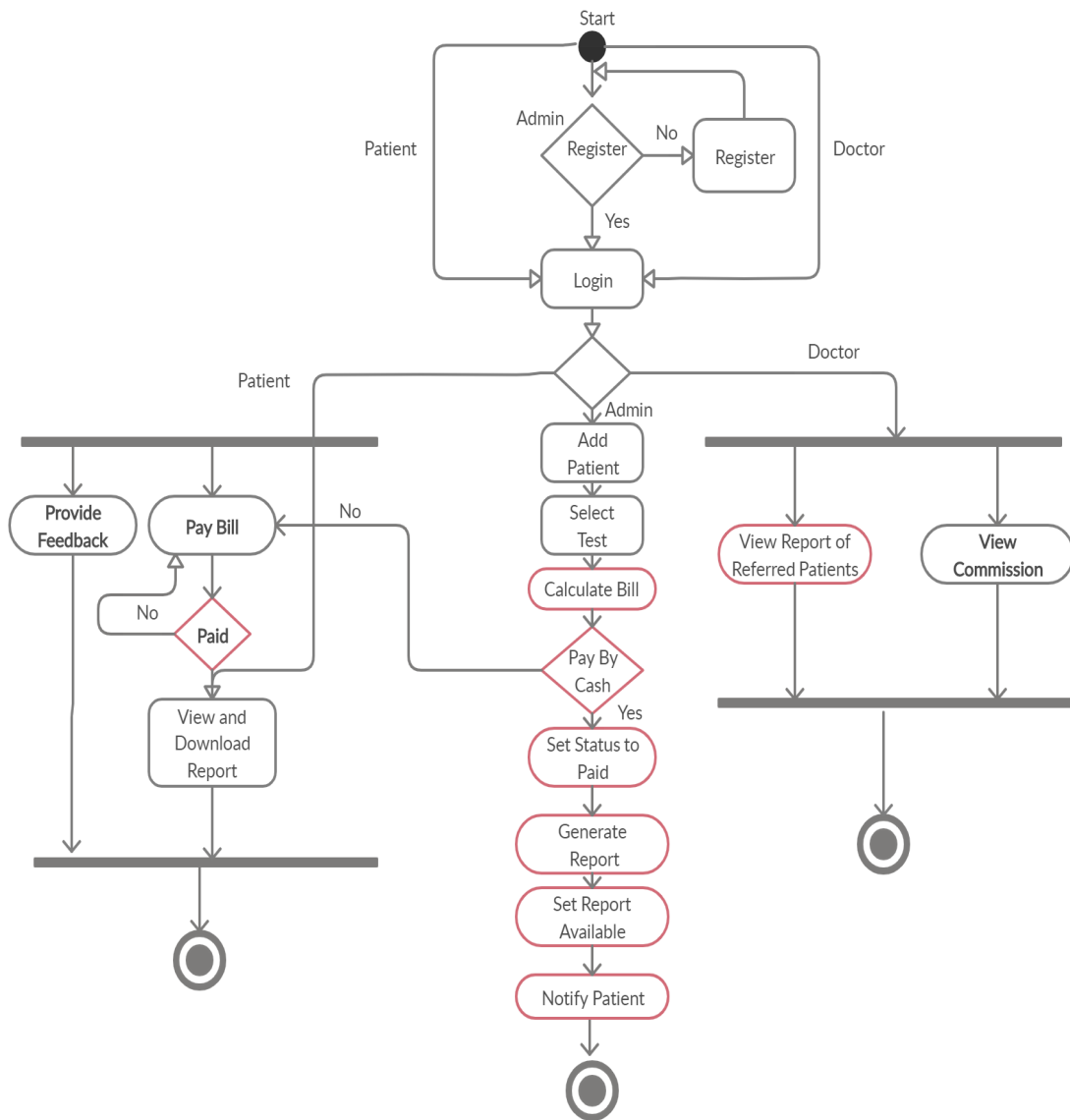


Figure 4.2: Activity Diagram

4.3 Entity Relationship Diagram

Entity-Relationship Diagram (ERD) illustrates the entities in database and the relationship between those entities. Following is the ERD of the Automated Biological Lab system in which all the entities, their relationships, and the cardinality (occurrence of relationship) of each entity is described in the figure 4.3.

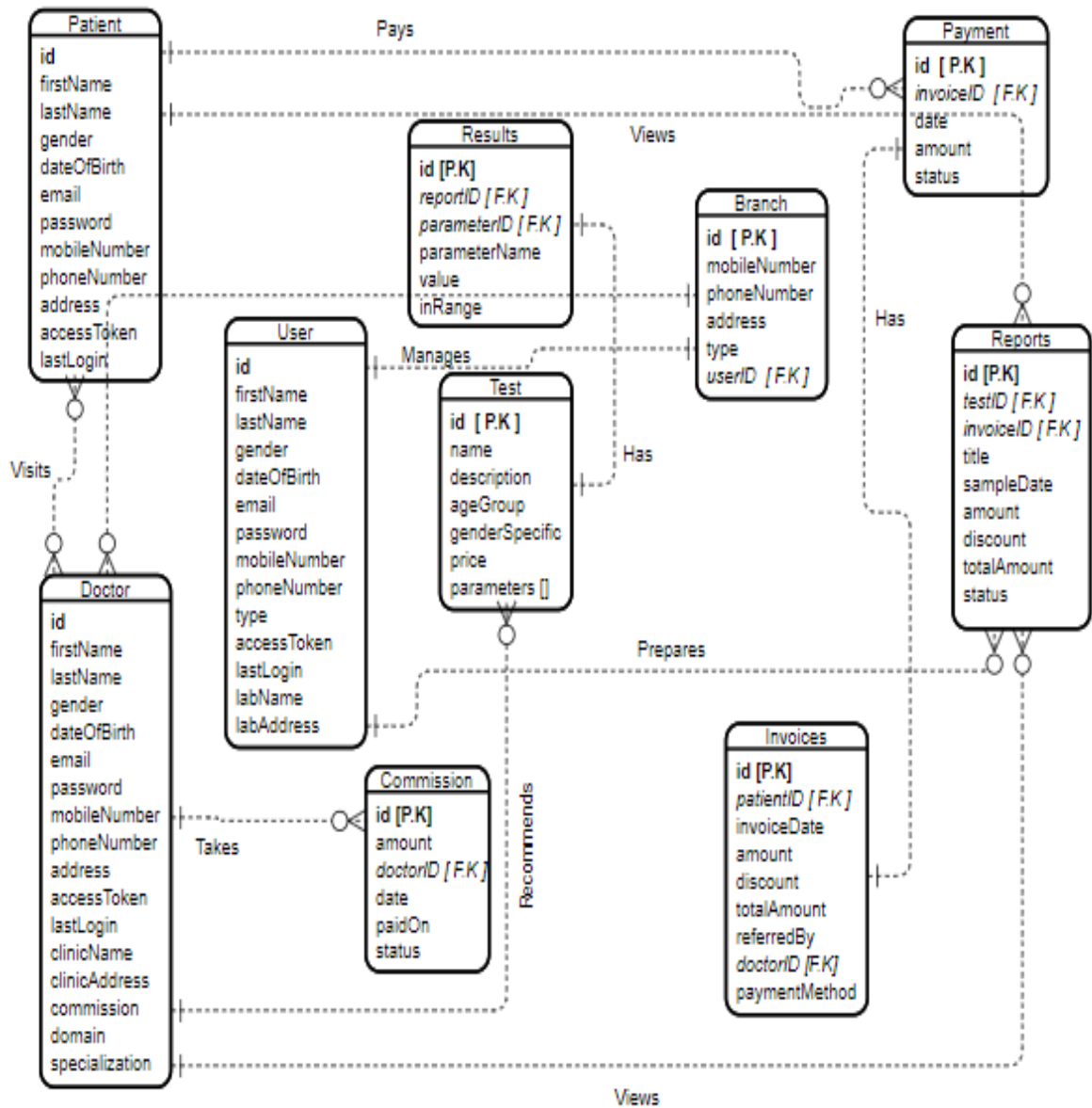


Figure 4.3: Entity Relationship Diagram

4.4 Sequence Diagram

Following is the sequence diagram of the functions that are to be performed when a patient interacts with the system is shown in the figure 4.4.

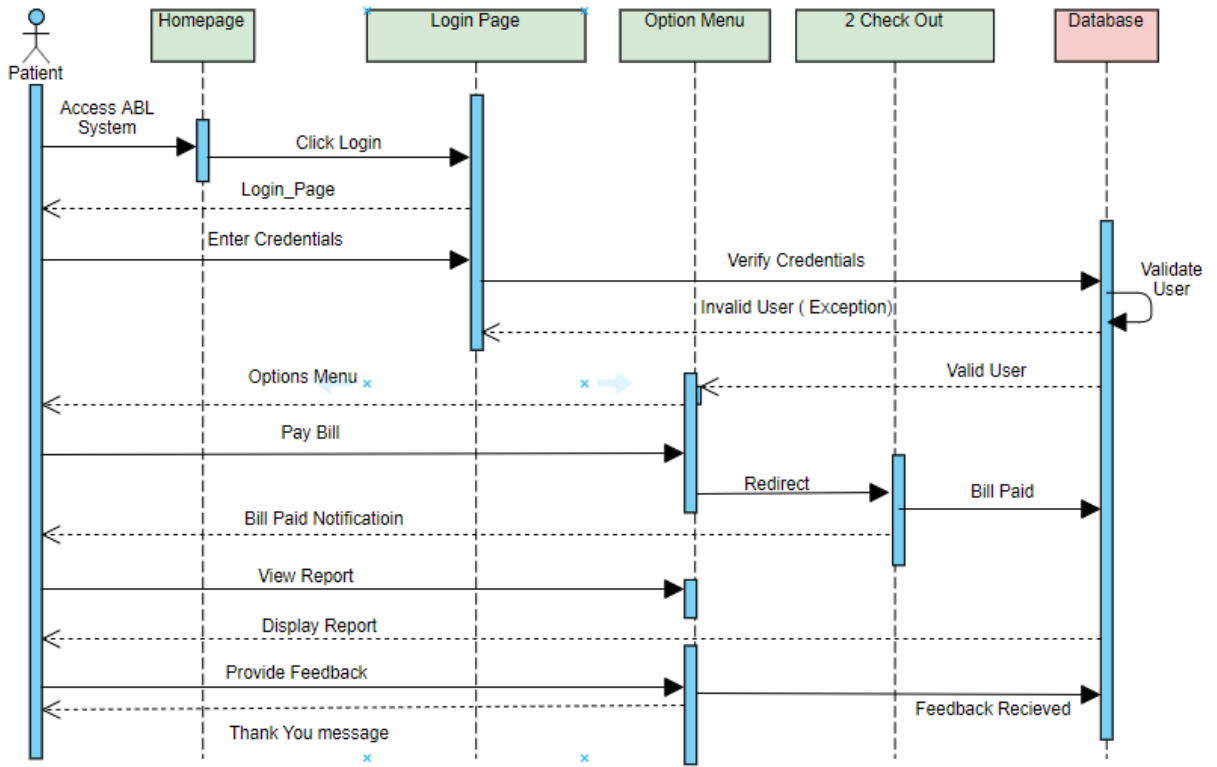


Figure 4.4: Sequence Diagram

4.5 Graphical User Interface

4.5.1 Main Screen



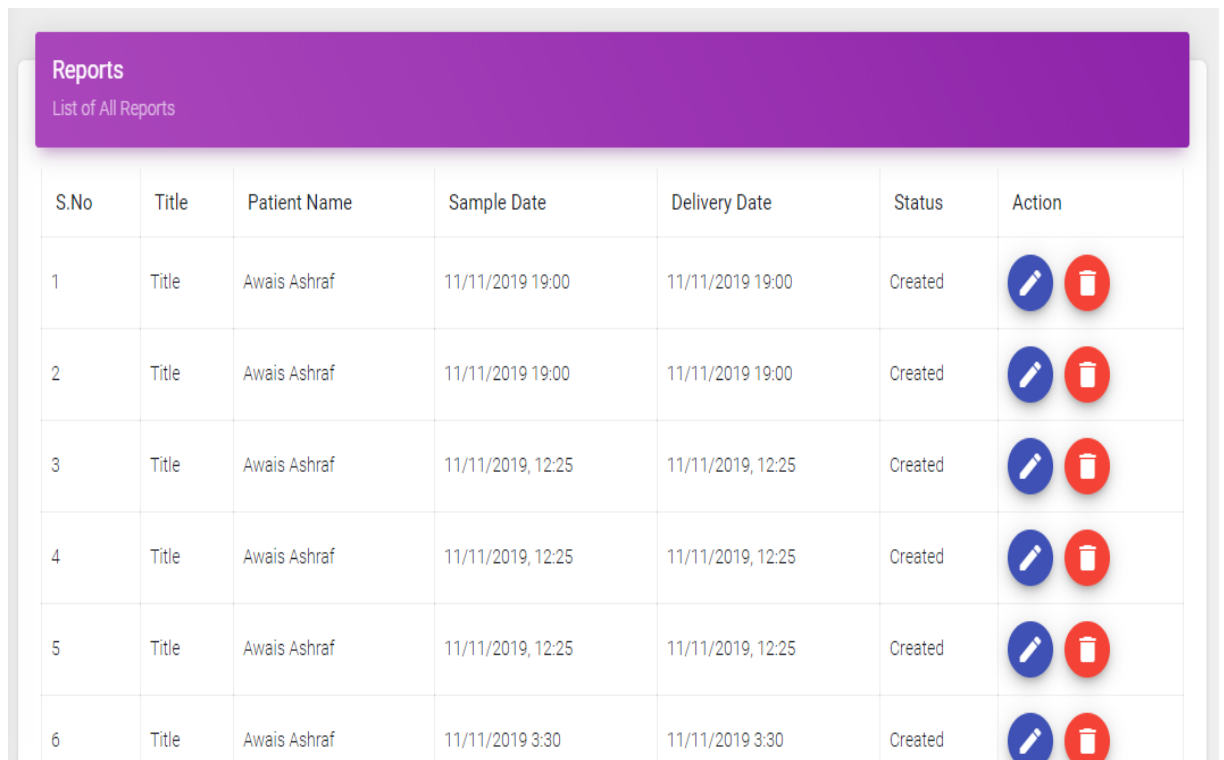
Figure 4.5: Main Screen

4.5.2 Add Test

The screenshot shows a web form titled "Add Test" with the subtitle "Complete Test Details". The form is contained within a purple header bar. Below the header, there are several input fields: "Test name *" (required), "For" (a dropdown menu), "Age Group" (a dropdown menu), and "Test price *" (required). Below these is a "Test duration *" field. A section titled "Test Parameters" contains two rows of input fields: "Parameter Name *" (required), "Unit *" (required), and "referenceValue *" (required). At the bottom left of the form is a link labeled "Add Parameter", and at the bottom center is a purple "SUBMIT" button.

Figure 4.6: Add Test

4.5.3 Reports List















S.No	Title	Patient Name	Sample Date	Delivery Date	Status	Action
1	Title	Awais Ashraf	11/11/2019 19:00	11/11/2019 19:00	Created	 
2	Title	Awais Ashraf	11/11/2019 19:00	11/11/2019 19:00	Created	 
3	Title	Awais Ashraf	11/11/2019, 12:25	11/11/2019, 12:25	Created	 
4	Title	Awais Ashraf	11/11/2019, 12:25	11/11/2019, 12:25	Created	 
5	Title	Awais Ashraf	11/11/2019, 12:25	11/11/2019, 12:25	Created	 
6	Title	Awais Ashraf	11/11/2019 3:30	11/11/2019 3:30	Created	 

Figure 4.7: Report Lists

4.5.4 Add Invoice

Add Invoice

Complete Invoice Details

Mobile Number * First name * Last name * Age * Gender

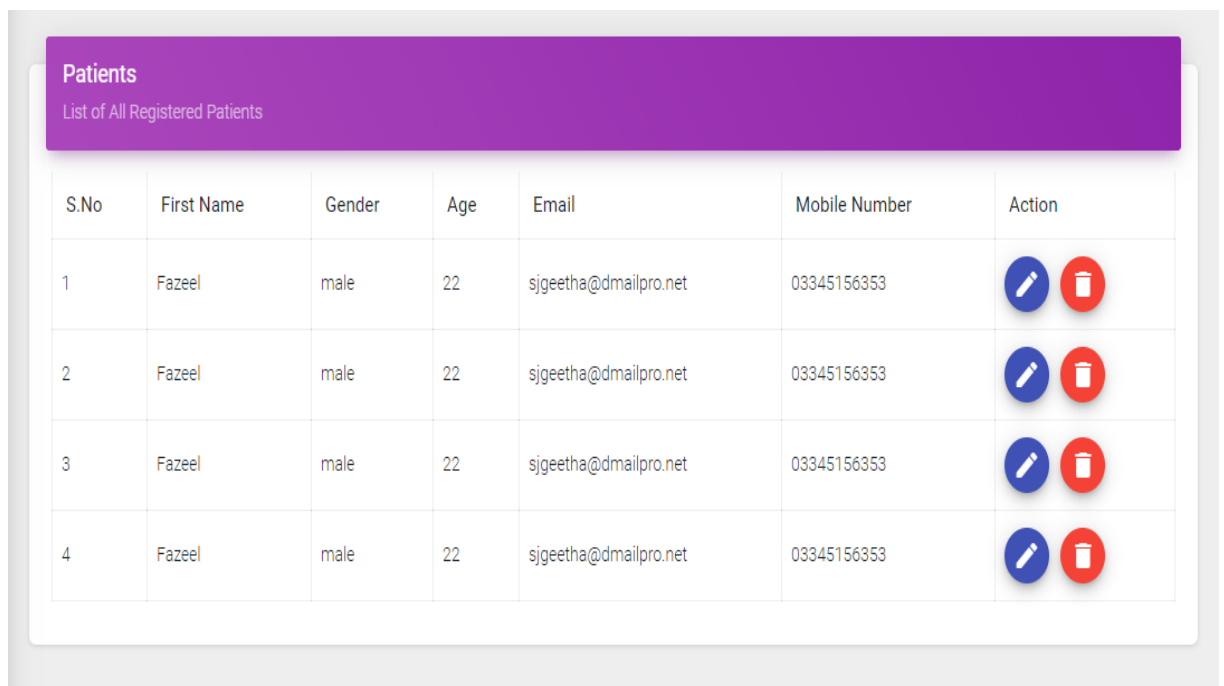
Doctors Tests

S.No	Name	Price
1	Blood CP	500
2	Hemoglobin	700
3	LFT	800

Total: 2000

Figure 4.8: Add Invoice

4.5.5 List of Patients











S.No	First Name	Gender	Age	Email	Mobile Number	Action
1	Fazeel	male	22	sjgeetha@dmilpro.net	03345156353	 
2	Fazeel	male	22	sjgeetha@dmilpro.net	03345156353	 
3	Fazeel	male	22	sjgeetha@dmilpro.net	03345156353	 
4	Fazeel	male	22	sjgeetha@dmilpro.net	03345156353	 

Figure 4.9: List of Patients

4.5.6 Payment Page of 2 CheckOut

Secure Checkout

✓ Review Cart

2 Billing Information

Country *

Full Name *

Address 1 *

Address 2

City *

State/Province

Postal Code

Phone

Email Address *

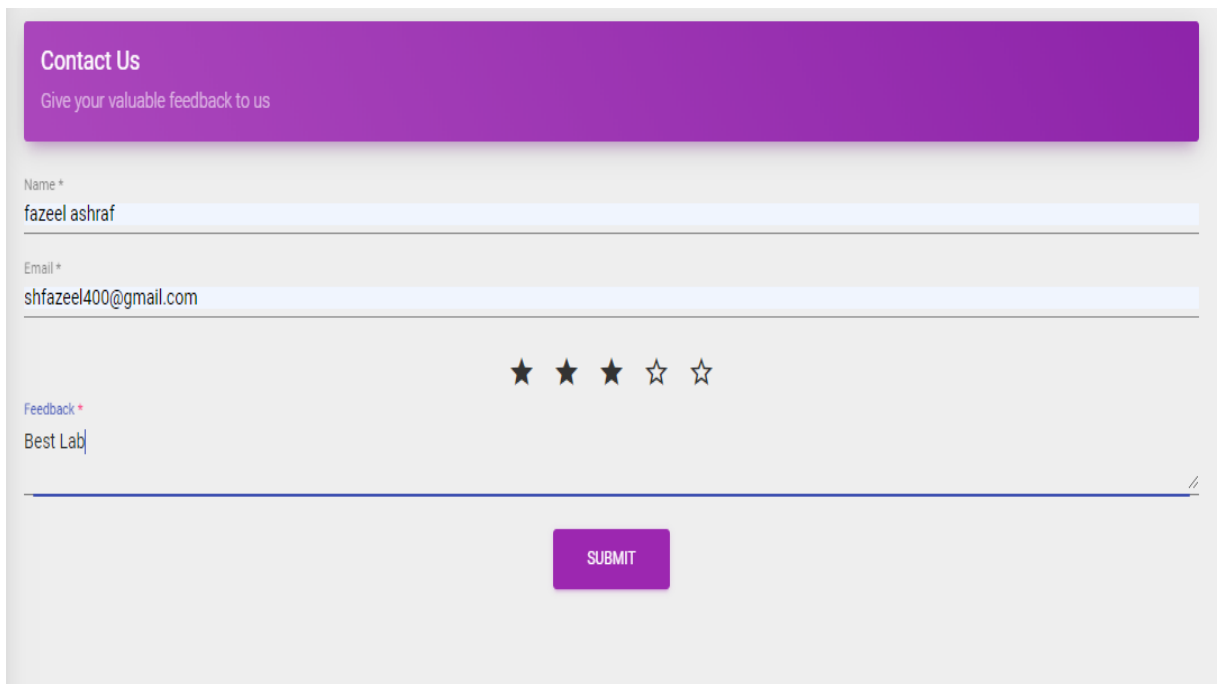
Upon completion of your purchase, a receipt will be sent to this e-mail address.

[Continue to Payment Method](#)

3 Payment Method

Figure 4.10: Online Payment Page

4.5.7 Reviews And Rating



The image shows a 'Contact Us' form with a purple header. The header contains the text 'Contact Us' and 'Give your valuable feedback to us'. Below the header, there are three input fields: 'Name *' with the value 'fazeel ashraf', 'Email *' with the value 'shfazeel400@gmail.com', and 'Feedback *' with the value 'Best Lab'. Above the feedback field, there is a five-star rating system with three stars filled and two empty. A purple 'SUBMIT' button is located at the bottom of the form.

Figure 4.11: Contact Us

Chapter 5

System Implementation

This section is about the implementation of the specified requirements and the design. It describes the tools and technologies and implementation strategies used during the software coding.

5.1 Tools and Technologies

Following are the tools and technologies used in our project:

5.1.1 HTML

HTML stands for HyperText Markup Language. It has various elements and tags which are used to define the structure and the layout of the content of web pages.

5.1.2 CSS

CSS is the short form of Cascading Style Sheets and is used to specify the layout and the other attributes of the HTML elements such as display position, text font, color and so on.

5.1.3 Bootstrap

Bootstrap is a CSS framework that is used to make mobile-first web applications. It has built-in classes that we can include in our application. We don't have to write all the classes by ourselves.

5.1.4 JavaScript

JavaScript is a scripting language for creating web pages, used on the client-side. It is used to create robust web applications. Moreover, it provides special effects and graphics for web pages.

5.1.5 JQuery

Jquery is a JS library framework used to simplify the operations performed on HTML elements. Its main purpose is to make searching and traversal easier to use JavaScript language.

5.1.6 Visual Studio Code

The system is developed in a visual studio code tool to help visualize the system during development. In addition to this, this tool provides many extensions that help you while coding. It supports debugging of the application, git control, and GitLab, tons of extensions each has its own functionality.

5.1.7 Adobe Illustrator

Adobe illustrator is an editor for the images which provides a lot of functionalities to perform on the images, that can be installed on desktop or laptop. In this project, adobe illustrator is used for logo designing of the website.

5.1.8 GIT

Git is used for tracking changes in the source code during the development of software as it is a distributed version control system used for tracking changes in source code. Moreover, it provides flexibility and security to the developers also. In my project git is used to track the changes and to secure my code.

5.1.9 MEAN Stack

All four of these technologies combine to create a mean-stack development toolkit to develop web applications. As the mean stack is efficient in creating robust web applications from client to server and server to database, so I have used the mean stack for creating this web application project.

5.1.10 MongoDB

MongoDB is the No SQL database program and it follows key-value pair structure which is known as JSON. It is a cross-platform document-oriented database. In MongoDB, tables are known as collections and their architecture consists of collections and documents.

5.1.11 Express

Express is a framework of Node.js and it is open source. It is used for making applications back-end and API's. It has a lot of built-in functionalities which can be used in your application.

5.1.12 Angular

Angular is a JS-based open-source front-end web application framework which is developed by Google. It is basically a TypeScript based framework that is used to make dynamic web applications and have a lot of features that we can use.

5.1.13 NodeJS

NodeJS is used to write server-side scripting and making API using express framework for your web application.

5.2 Implementation Strategy

The implementation strategy is not very complex. As it is a web application so the project requires deep knowledge and skills of web development and full-stack development, so the latest technologies of nowadays and frameworks are used in this project.

- Operating System: Windows 10.
- Database Management: MongoDB.
- Front-End: Angular.
- Back-End: Node.JS.
- Development Server: Heroku.

5.3 Functionalities of the System

The following are the functionalities provided by the ABL to the users according to their domain.

- **Patient:** The patient can view his medical reports online, and can download the reports as well. He can edit his profile and can pay the bill online through 2 checkout payment method. Moreover, the patient can rate and review his experience with the lab.
- **Doctor:** Doctors who are registered with the ABL and refer the patients to the specific lab can view the medical reports of their referred patients and their commission for referring the patients as per agreement.
- **Admin:** Admin can add/register the patient and doctors in the system. Admin is responsible for calculating the bill, generating the invoice, and preparing the medical reports of the patients. Moreover, the admin can view the feedback of the patients about the lab.

Chapter 6

System Testing and Evaluation

Testing of the whole project is done by using the Manual Based Testing approach. Each component of the system has been tested manually by running it on the user end. The testing types, which are required by the system are written below:

6.1 Graphical User Interface Testing

The Graphical User Interface of the system is user-friendly and self-explanatory. All the content (elements and the objects) of the GUI has been placed using a pleasant color scheme. The font of UI labels is clearly readable and comply with the standards. The functionality of buttons against the click works properly according to the specifications. Moreover, alerts and error messages are displayed when some mandatory text field is left blank or the user enters the wrong input. Encrypted input is taken in the case of sensitive info such as passwords, etc. and all the design requirements of the UI of the system have been met.

6.2 Usability Testing

Every screen of the application is labeled with the instructions for the understanding of the user.

6.3 Software Performance Testing

6.3.1 Volume Testing

Free version of Heroku is used which provides the server of 512 MBs, so ABL can handle the data up to the capacity of the free version of Heroku.

6.3.2 Stress Testing

Stress testing is done by increasing the number of users than usual (according to the capacity of the server being used i.e. free version of Heroku), who are using the system at the same time. The applications work fine under such conditions.

6.4 Compatibility Testing

The ABL is a web application and can be used on any device browser like mobile, tablet, PC and desktop. The responsiveness of the application has been tested by running it on different devices such as laptops and mobile. It can automatically adjust the content and the layout according to the size of the device.

6.5 Exception Handling

In my application, a lot of checks are there at the back-end side when calling API's if some error occurs then the server sends you an error message. Built-in functions are also used e.g "ngNativeValidate".

6.6 Load Testing

Load testing is performed in our application when multiple users use it at one time. Heroku automatically synchronizes the data being entered into the application interface and sends it into the areas in which that data is required, the application ensures better efficiency and performance.

6.7 Security Testing

It has been ensured by verification that only authorized users can access the application. Whenever the wrong credentials are entered by the user for logging in to the system or incorrect input is provided, an error is generated and further access to the system is denied. No one has the right to access from the url until he/she is verified and logged in.

6.8 Installation Testing

Installation testing has been done by uploading and using the application on the server of the free version of Heroku. It works correctly, as per the specified requirements.

6.9 Test Cases

Following are the test cases of our application.

6.9.1 Login Test Case

This test case basically checks the validity of the credentials of the user. The test case is successful and allows the user to access the system functionalities when the username and password entered by user are correct. The test case fails when the entered credentials are wrong.

Table 6.1: Test Case: Login

Test Case ID	01	
Test Description	Logging in with existing admin/patient/doctor account.	
Revision History	Created by Muhammad Fazeel Ashraf on 05-11-19.	
Function to be tested	Login an account.	
Test Setup	Internet connectivity.	
Test Execution	Expected Result	Actual Result
Select Login option from the home page of the application.	Login page/window should be opened.	Login window has been opened successfully.
Enter a valid username and password.	The system should validate and accept these credentials.	The entered credentials have been accepted by the system successfully.
Click Login button to access the system functionalities.	User should be able to login to the system.	User has been successfully logged in to the system.

6.9.2 Add New Patient Test Case

This test case works when a new patient's record is added in the database record of the system. The test case is successful when the data entered by the user matches with the constraints of the required text and fails when the requirements of the respective function are no full filled.

Table 6.2: Test Case: Add New Patient

Test Case ID	02	
Test Description	Adding a new patient record in the system database.	
Revision History	Created by M Fazeel Ashraf on 05-09-19.	
Function to be tested	Add new patient.	
Test Setup	Internet connectivity.	
Test Execution	Expected Result	Actual Result
Select Add New Patient option from the option menu.	A new window should be opened containing patient record form.	Window containing patient record form has been opened successfully.
Enter details of the patient (i.e. Name, Date of Birth, gender, email, contact number, address, password etc.)	Patient profile should be created and all the details should be recorded in the database system.	Patient profile has been created and patient record has been saved in the database system successfully.

6.9.3 Sign-Up A New Admin Test Case

This use case works when a new user account is created. The test case is successful when all the requirements for signing up are full filled and fails otherwise.

Table 6.3: Test Case: Sign-Up A New Admin

Test Case ID	03	
Test Description	Registration/Sign up of a new patient.	
Revision History	Created by M Fazeel Ashraf on 05-11-19.	
Function to be tested	Register/Sign up a new patient.	
Test Setup	Internet connectivity.	
Test Execution	Expected Result	Actual Result
Select Register Now/Sign up option form the option menu.	Sign up screen should be opened.	Sign up screen has been opened successfully.
Add the credentials of the patient.	System should add these credentials in the database.	All the credentials have been saved successfully in the system database.
Click on Sign up button to create an account.	A new patient account should be created.	A new account for the patient has been created successfully.

6.9.4 Select Medical Tests Test Case

This test case works when the user selects medical tests from the test list.

Table 6.4: Test Case: Select Medical Tests

Test Case ID	04	
Test Description	Selecting medical test from the list of medical tests.	
Revision History	Created by M Fazeel Ashraf on 05-12-19.	
Function to be tested	Select Medical Tests.	
Test Setup	Internet connectivity.	
Test Execution	Expected Result	Actual Result
Select the medical test (recommended by the doctor) from the drop down menu/ list of medical test.	Tests selected by the user should be checked.	Tests selected by the user have been checked successfully.
Click on the confirm button.	An invoice should be generated containing the patient id, other patient details and the selected medical tests.	An invoice has been generated successfully containing the patient id, other patient details and the selected medical tests.

6.9.5 Calculate Bill

This test case works when the bill is calculated for the selected tests.

Table 6.5: Test Case: Calculate Bill

Test Case ID	05	
Test Description	Calculating bill for the medical tests.	
Revision History	Created by M Fazeel Ashraf on 05-13-19.	
Function to be tested	Calculate Bill.	
Test Setup	Internet connectivity.	
Test Execution	Expected Result	Actual Result
Click on calculate bill option from the menu.	A calculate bill window with the name of all the selected medical tests and their price should be popped up by the system.	A window with the name of all the selected medical tests and their price has been popped up by the system successfully.
Click on continue button from the popped up calculate bill window.	Price of the selected tests should be added, total bill should be calculated, and the commission of the doctor should be calculated according to the total bill.	Price of the selected tests had been added, total bill has been calculated, and the commission of the doctor has also been calculated successfully, according to the total bill.

6.9.6 Bill Payment Test Case

This test case works when user pays the bill. The test case is successful when the information entered by the user is correct and is unsuccessful when any of the information provided by the user is wrong.

Table 6.6: Test Case: Bill Payment

Test Case ID	06	
Test Description	Paying the amount of bill through 2 checkout.	
Revision History	Created by M Fazeel Ashraf on 05-14-19.	
Function to be tested	Payment of bill.	
Test Setup	Internet connectivity.	
Test Execution	Expected Result	Actual Result
Select Pay Bill option from the option menu.	The window should be redirected to the 2 checkout payment window.	The window has been redirected to the 2 checkout payment window successfully.
Enter the required information (i.e. Credit card number, card holder name, Card pin, contact number etc.)	The entered information should be accepted by the system.	The entered information has been accepted by the system successfully.
Click on Continue/Pay button.	The entered credentials should be validated, the transaction of calculated bill amount should be performed and the receipt should be generated.	The entered credentials have been validated, the transaction of calculated bill amount has been performed and the receipt has been generated by the system successfully.

6.9.7 Feedback Test Case

This test case works when the user provides feedback about his experience with the respective biological lab.

Table 6.7: Test Case: Feedback

Test Case ID	07	
Test Description	Providing feedback about the biological lab by the patient.	
Revision History	Created by M Fazeel Ashraf on 05-15-19.	
Function to be tested	Feedback about the lab.	
Test Setup	Internet connectivity.	
Test Execution	Expected Result	Actual Result
Select feedback option from the nav bar.	A window containing stars for rating and text area to write about your experience with the lab should be opened.	A window containing stars for rating and text area to write about your experience with the lab has been opened successfully.
Click on the Confirm or OK button.	The reviews should be submitted and the feedback window should be closed.	The reviews have been submitted and the feedback window has been closed successfully.

6.9.8 Generate Report Test Case

This test case works when the lab in-charge/lab technician generates the report of biological test performed.

Table 6.8: Test Case: Generate Report

Test Case ID:	08	
Test Description	Preparing the report for the medical tests performed on the blood sample of the patient by the lab technician/pathologist	
Revision History	Created by M Fazeel Ashraf on 05-16-19.	
Function to be tested	Generate report of medical test.	
Test Setup	Internet connectivity.	
Test Execution	Expected Result	Actual Result
Select an invoice from the list of pending invoices.	Selected invoice should be opened in a new window.	Selected invoice has been opened in a new window.
Click on Generate Report button from the opened invoice window.	The report template page should be opened containing the reference values for the medical tests according to the invoice. (Result of each test should be on new page).	Report template page has been opened containing the reference values for the medical tests according to the invoice (a new page for each test).
Enter the obtained values in the field of actual values.	Entered values should be accepted by the system.	Entered values should be accepted by the system.
Click on OK button.	Reports should be saved in the pdf format, the invoice should be removed from the pending invoices list, and a notification should be sent to the respective patient via email.	Reports have been saved in the pdf format, the invoice has been removed from the pending invoices list, and a notification has been sent to the respective patient via email successfully.

6.9.9 View Report Test Case

The test case is successful when the report is prepared and the status of report is set to available.

Table 6.9: Test Case: View Report

Test Case ID	09	
Test Description	Viewing the report for the medical tests performed on the blood sample of the patient.	
Revision History	Created by M Fazeel Ashraf on 05-17-19.	
Function to be tested	View report of medical test.	
Test Setup	Internet connectivity.	
Test Execution	Expected Result	Actual Result
Click on view report option from the option menu.	View report window should be opened, containing all the previous and the current reports of the patient from our lab.	View report window should be opened, containing all the previous and the current reports of the patient from our lab.
Click on the report form the list that you want to view.	Selected report should be opened successfully.	Selected report has been opened successfully.

Chapter 7

Conclusions

While implementing this system I have learnt a lot of new technologies and gained knowledge of various emerging technologies. Initially, I faced a lot of difficulties regarding Mean Stack development. All the knowledge which I have learned regarding this project from Udemy lectures, by asking my supervisor and from my previous concepts as well. I have learned the required knowledge after some time. This technology provides a lot of built-in functionalities and are way better than previous technologies which are used to minimize the complexity of problem description. ABL is my first application in Mean Stack which is complete. It has a way better interface as compared to the existing systems. I have tested the application from the user's end by asking my colleagues to test my application. This application is efficient in the way it reduces the tiring work done by the lab technicians, and the doctor commission is also calculated when the patient pays the amount. Alerts are also generated to inform the user that the system is working fine.

7.1 Future Work

Enhancements and bringing new concepts in the software is an unending process, however I am planning to add following functionalities in the future.

- Stock management system to maintain the quality and quantity of the chemicals needed for the lab instruments. The system will automatically check the level of required chemicals according to the date of the recent refill and will notify the technicians[3].
- Patients will be able to give the samples of the simple tests while sitting at home (like urine and blood tests)[4]. The patient will select the tests, will enter his address, and depending upon the requirements of the test, the lab technician will come to the

mentioned address and take the samples to the lab. When reports will be prepared, the patient will be notified via email , and he can view his reports by signing in.

References

- [1] Tru. Solutions. Limit LIS, July 2015. <http://elab.in>. Cited on p. 6.
- [2] ELAB, January 2013. <https://www.ruro.com/limitlis/tour>. Cited on p. 6.
- [3] Excel Lab Pvt Ltd. Excel Labs, November 2012. <https://www.excel-labs.com/Home>. Cited on p. 43.
- [4] Islamabad Diagnostic Center Pvt Ltd. IDC, August 2016. <https://idc.net.pk/>. Cited on p. 43.