American International University-Bangladesh (AIUB)

Department of Computer Science

Faculty of Science &Technology (FST)

Spring 22 23

Section: A

Software Quality Assurance and Testing

VojonBilash

A Report submitted
By

SN	Student Name	Student ID
1	Faiza Tasnim	19-41552-3
2	Md. Alinur Hossain	20-41855-1
3	Sadia Afrin Sara	20-41834-1
4	Sharia Tasnim Adrita	20-41895-1

Under the supervision of

ABHIJIT BHOWMIK

Associate Professor

And

Special Assistant – Office of Student Affairs

Department of Computer Science

American International University-Bangladesh

Software Test Plan

for

VojonBilash

Version 1.0 approved

Prepared by

Faiza Tasnim
Md. Alinur Hossain
Sadia Afrin Sara
Sharia Tasnim Adrita

AIUB

<13 April,2023>

Checked By Industry Personnel

Name:
Designation:
Company:
Sign:
Date:

Table of Contents

Revision History

Revision	Date	Updated by	Update Comments
0.1	2023.04.08	Md. Alinur Hossain	First Draft
0.2	2023.04.09	Sadia Afrin Sara	Second Draft
0.3	2023.04.10	Faiza Tasnim	Third Draft
0.4	2023.04.11	Sharia Tasnim Adrita	Fourth Draft

1. TEST PLAN IDENTIFIER:RS-MTP01.3

2. REFERENCES

- Software Quality and Testing Course PowerPoint Slides
- Software Requirement documentation
- Software Requirements Specification (SRS) Document
- https://www.w3schools.com/

3. INTRODUCTION

Background to the Problem

As we can see physically bringing food from the restaurant or having food by dine in is a problem for so many people. Many people now a days prefer the food delivery option over dine in. Sometime people do not have a time or mood to go to the restaurant. Also, there is a transportation problem. Some people want to eat from a restaurant which is very far from their home but they did not want to travel. There are also have bad weather sometimes on that time people does not want to go outside but want food in their home.

Solution to the Problem

People use Food panda, Uber Eats, HungryNaki etc. services to get their desired food from desired vendor at home. That's exactly how our VojonBilash works like Food Panda. VojonBilash is an E-cafe management system where we provide food delivery service through user and restaurant interaction. An E-Cafe Management System helps conserve time by taking order through online platform, preparing and delivering in a swift manner. The main purpose of this project was to automate the process of ordering to receiving the food. The rising demand of food delivering culture that delivers food at the customer's doorstep complements the intention of this project. It will be very helpful for the users because they will be able to all types of restaurants in one place. This is a profitable business for both the restaurants and the service provider. Similar to the food delivery services we provide direct interaction between customer and the restaurant and whole process is being monitored by the service management. From the customer, the order is sent to the deliveryman to pick up food from the restaurant. This process is managed through the website. After picking up the food deliveryman delivers the food to the address and takes payment from the users via online payment or Cash on Delivery. In the world of globalization, every service is internet based. Though it is a web-based application, for security purposes, there will be an option for users which is to log in. Also, users can see the features of their Personal details, Update profiles, Choose

restaurants, and Order food, Confirm foods by logging in to our system. Also, in the future, we will make further changes to this project's login and user security system and also try to implement some features for customers.

4. REQUEIREMNT SPECIFICATION

4.1 System Features

1. Customer

1.1 Registration: Users have to register with our system when they decide to use the system.

1.2 Login: After successfully registering to our system, users will have to login to use the system.

1.3 Logout: The user can logout from the system anytime.

1.4 Change Password: Users can change their password if needed and the old password will be required for security at that time.

1.5 View Restaurant: User can see the restaurant

1.6 Choose Restaurant: User can choose the restaurants

1.7 Order Food: Users can order food through the system.

1.8 Edit/Change user information: User can update/edit information in their profile if needed.

1.9 Confirm foods: Users can confirm their order through our system.

Priority Level: High

Precondition: user must have a valid user name and password

4.2 System Quality Attributes

USABILITY:

Since everyone is the main stakeholder of this system, everyone can use it and benefit from it. As there can be many users for this system we have set our user login timeout. The features of this system are very easy to learn and beginners can easily learn to use the system.

RELIABILITY AND CORRECTNESS:

Complete testing is needed to make the software completely bug-free, but it is impossible to completely eliminate all bugs in software through testing, it is necessary to perform thorough testing to minimize them. However, this testing process is time-consuming. But our software can give output as the user wants. Despite this, our software has been extensively tested and verified to meet the user's requirements without generating any false output. The system's security has also been guaranteed, making it reliable for everyday use by users.

MODULARITY:

To develop the system, we adopted the MVC (model-view-controller) format, which allowed us to build the system in modules. By using this approach, we were able to easily identify and fix bugs in individual modules without affecting the other modules. This made the testing process smoother and more efficient. Additionally, the modular structure enabled us to easily incorporate new features into the system as needed.

MAINTAINABILITY:

Maintainability refers to how effective the maintenance team is and the ease with which they can carry out their duties, which include resolving identified errors, adding new features, or modifying existing ones. Because the software is designed using modules, the team can make modifications during the development phase in an equally efficient manner.

EFFICIENCY:

Efficiency is a crucial characteristic of a system and is gauged by the duration it takes to accomplish assigned tasks. Given that this application will be used extensively by end-users, we have made a concerted effort to ensure it operates at a high level of efficiency. With customers being the primary stakeholders, optimizing efficiency has been our primary concern.

TESTABILITY:

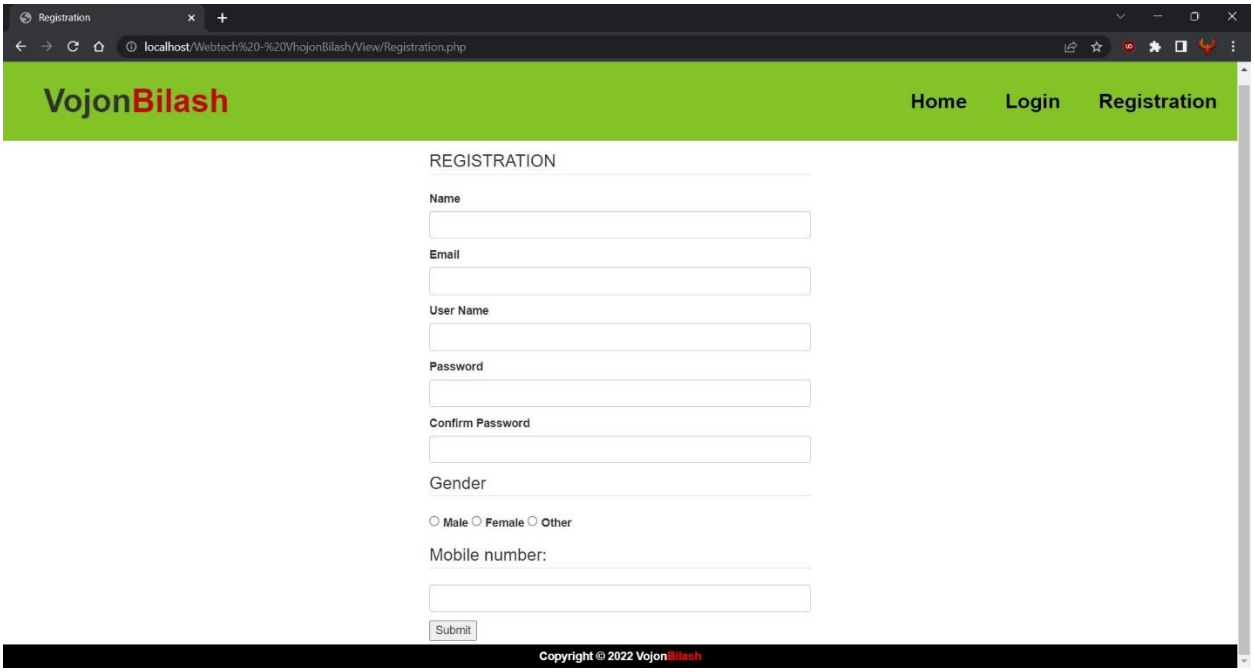
Testability refers to the ease with which the testing team can carry out their tasks. Testability is influenced by the modularity of the system. Our system was developed using the MVC (model-view-controller) format, which follows a modular approach. This allows the testers to easily test each module individually, rather than having to go through all the modules to identify and fix any issues. The modular structure of the project therefore enables the testers to fix any bugs in individual modules with ease.

FLEXIBILITY:

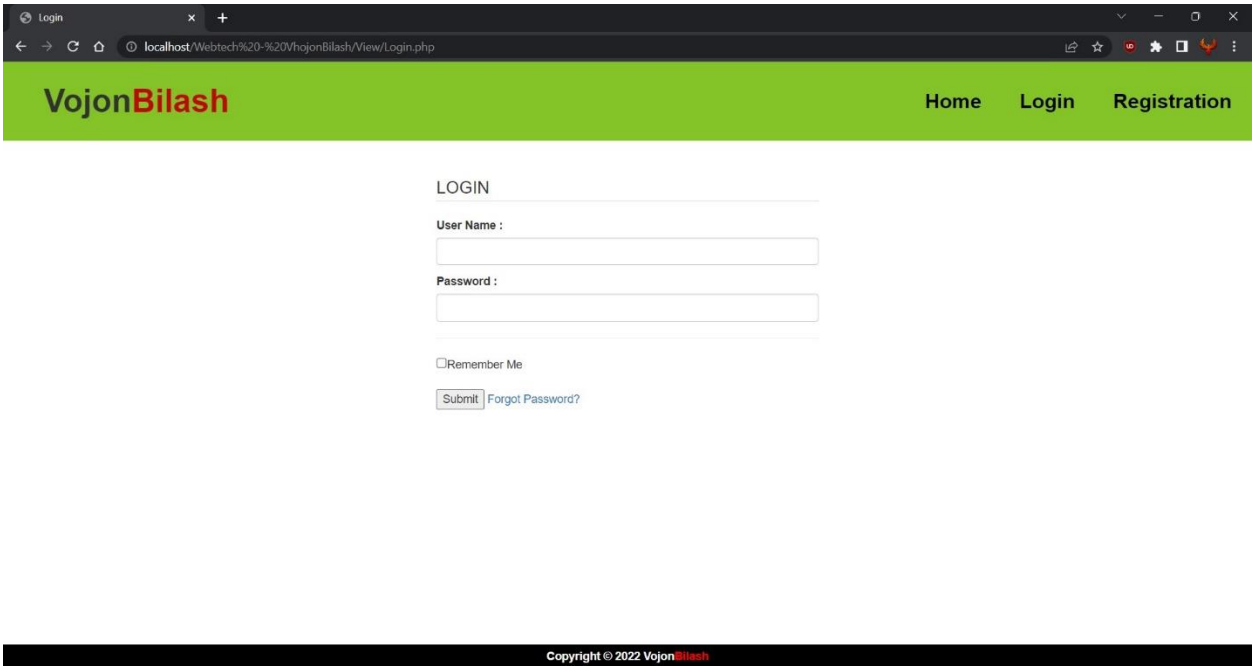
Flexibility refers to the level of difficulty involved in making changes to a program in operation. If a maintenance programmer has been supporting the product for at least eight months, they should be capable of creating a new copy output for the product, which includes modifying the code and conducting tests, in no more than two hours of work.

4.3 System Interface

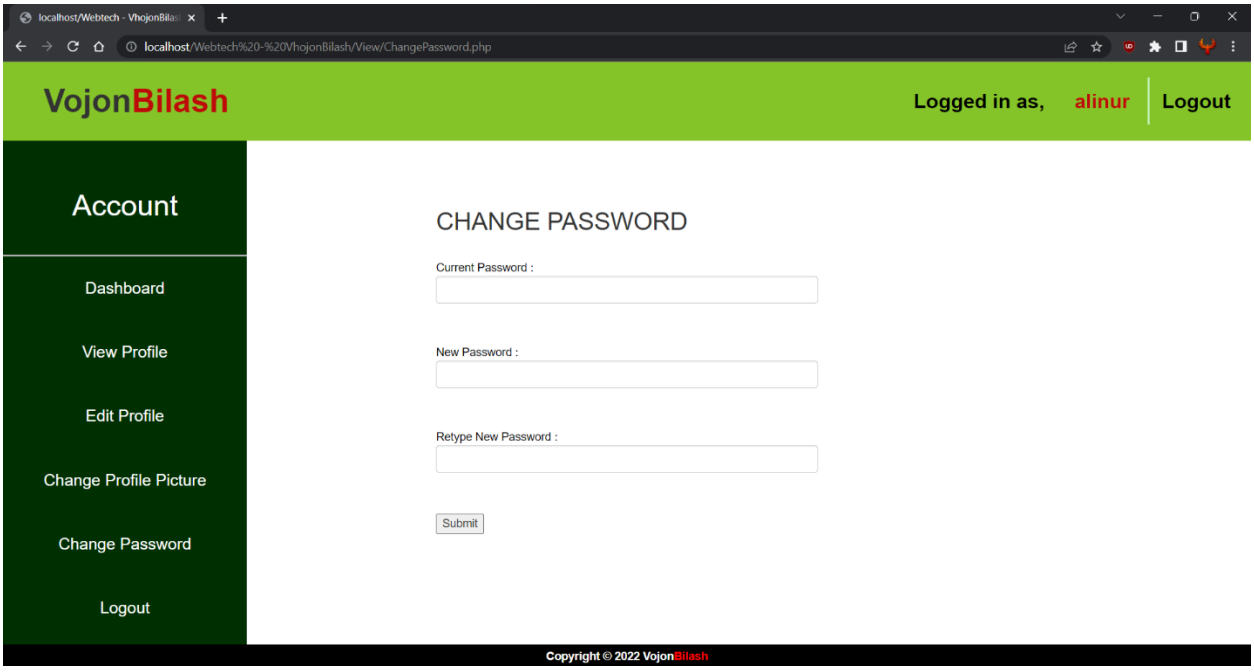
1. Users can register or signup here.



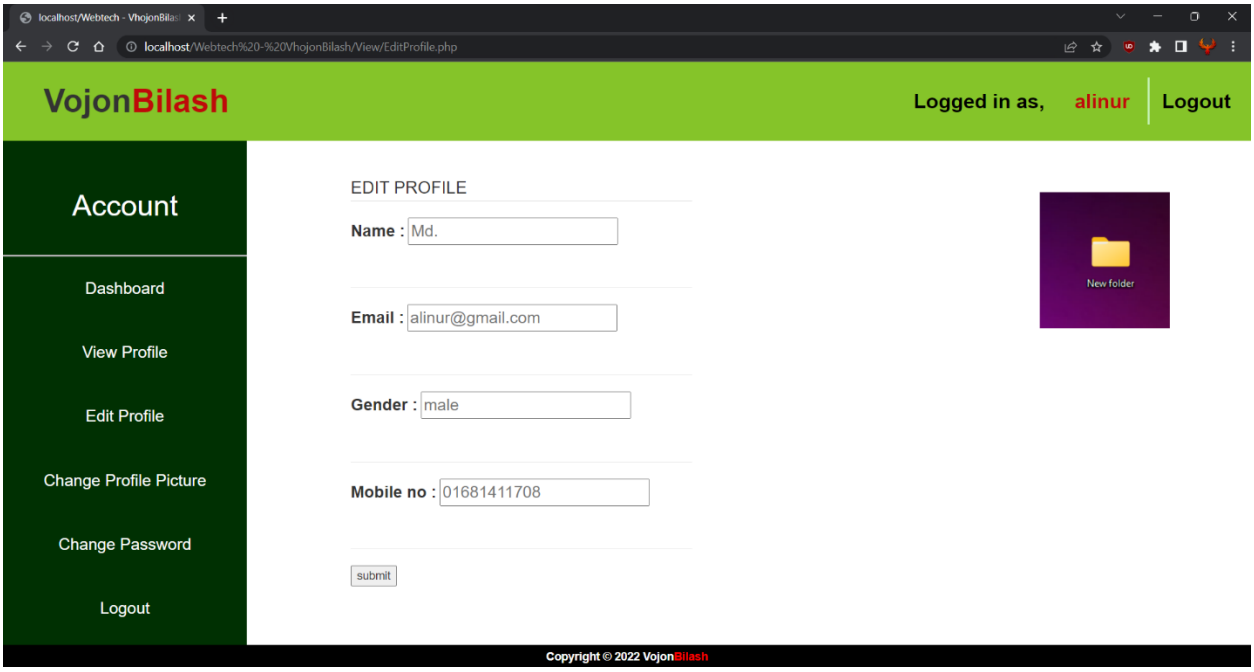
2. Users can login from this page.



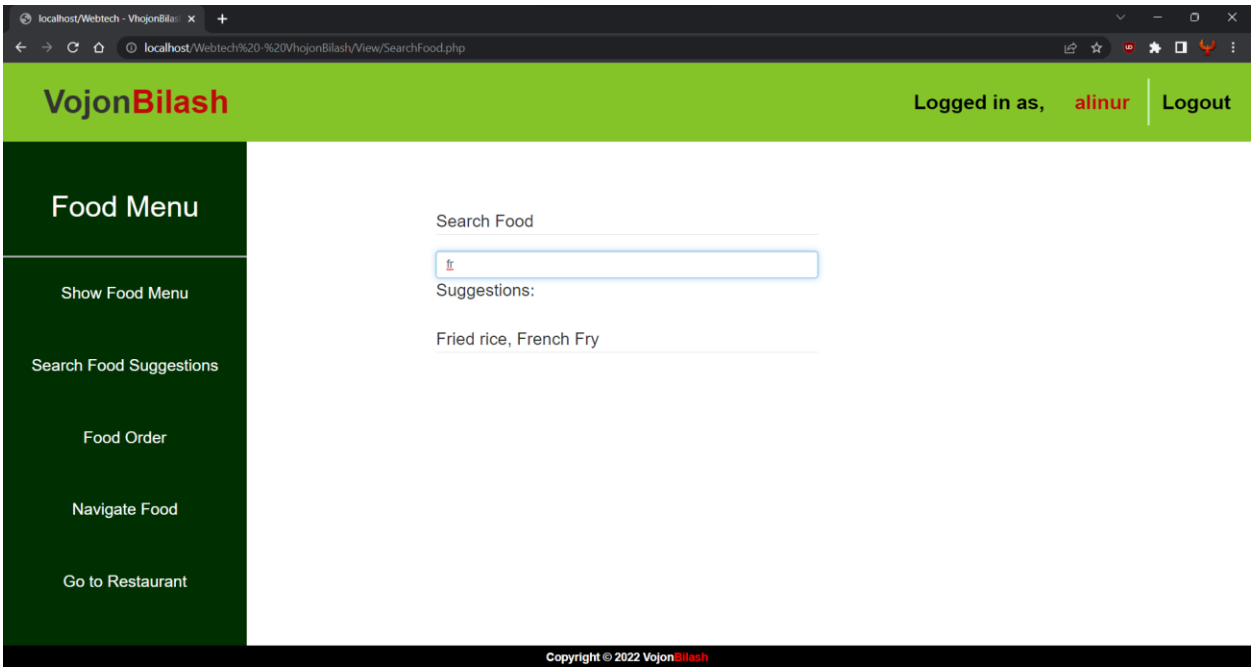
3. Users can change password.



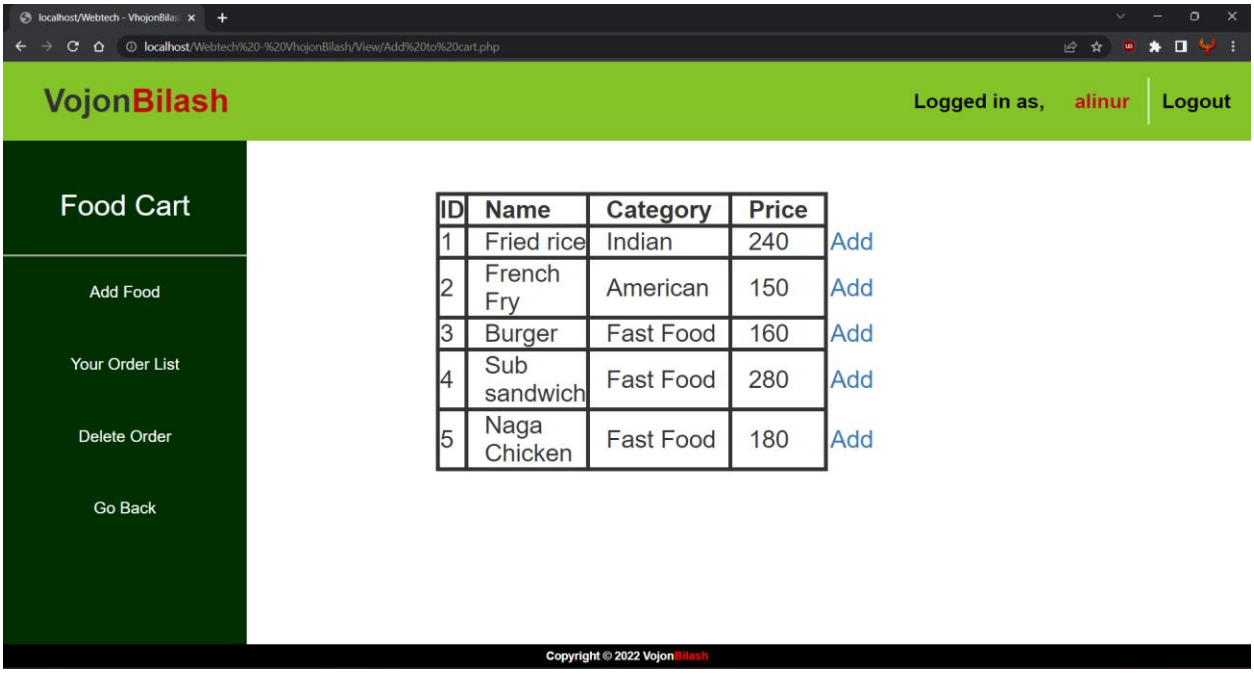
4. Profile can be edited from this page.



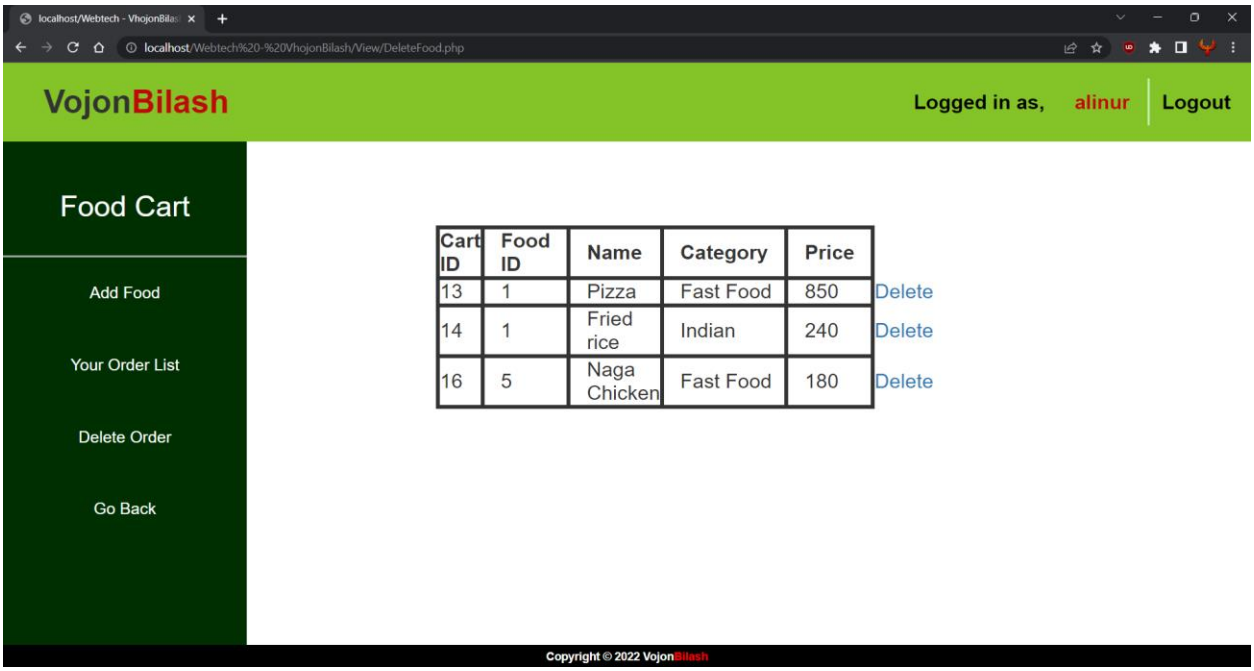
5. Food searching is available also.



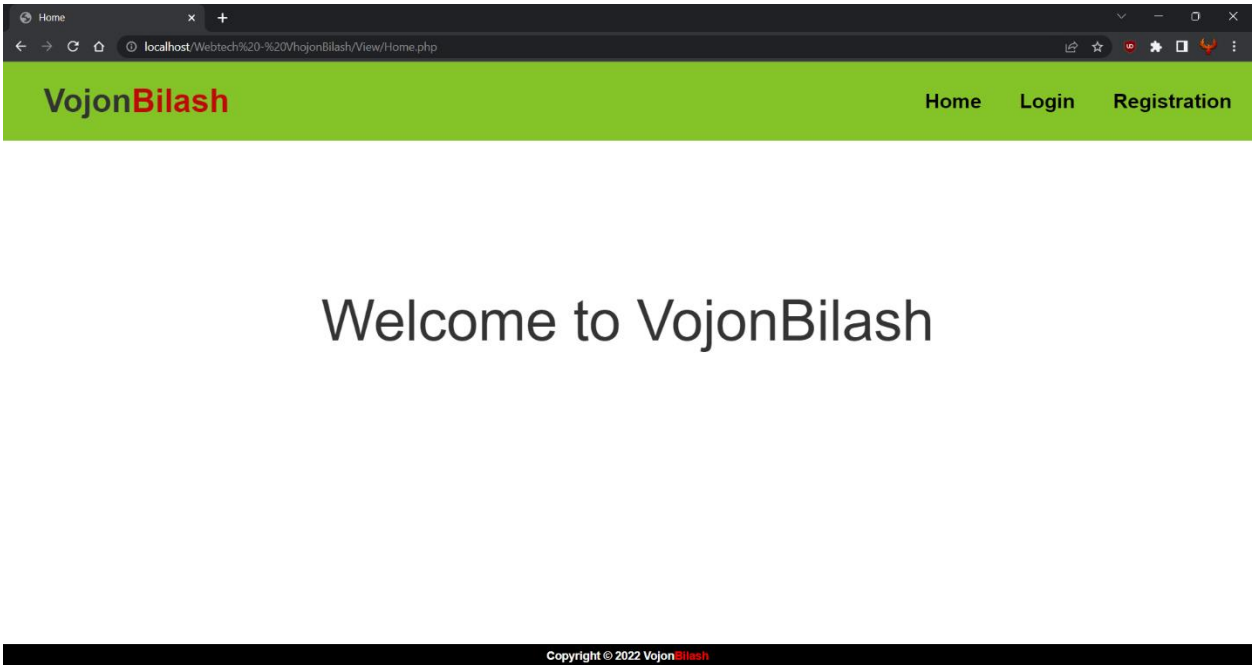
6. Customer can order their food.



7. Customer also can delete their order.



8. At last, user can successfully logout from their account.



4.4 Project Requirements

Effort Estimation:

We used COCOMO model for project estimation.

- **Project type: Semi-detached**
- **SLOC = 10000**
- **Coefficient<effort factor> =3**
- **P= 1.12**
- **Effort= PM= Coefficient<effort factor> *(SLOC/1000) ^P**

  =3*(10000/1000) ^1.12 =39.54
- **Development Time= DM= 2.50*(PM)^T**

  =2.50*(39.54) ^0.35=9.05
- **Required number of people =ST= PM/DM = 39.54/9.05=4.36**

Budgeting:

- **Development salary 9 month: Considering 600 taka per developer hourly**
  - **600*1584=950400 taka**
- **Requirement analysis: Hourly=400 taka**
  - **Analysis expense=400*176=70400 taka**
- **Transportation cost: 15500 takas**
- **Training and hardware cost: 150000 takas**
- **Rent expense: 18000*9=162000 taka**
- **Total utilities: 60000 takas**
- **Maintenance cost: Expense per hour =1500 taka**
  - **Extra hour needs for maintenance 80 hours**
  - **Total maintenance = 80 * 1500=120000 taka**
- **Other human resource: Project manager = 1* 28000*9=252000 taka**
  - **Accountant = 1* 22000*9=198000 taka**
  - **Technical staff = 2* 15000*9=270000 taka**
  - **Total cost= 252000+198000 +270000=720000 taka**
  -
- **Total estimate expense: 950400+70400+15500+150000+162000+60000+120000+720000=2248300 taka**
- **Profit: 2248300*25%= 562075 taka**
- **Final project budget: 2248300+562075=2810375 taka**

5. FEATURES NOT TO BE TESTED

The following is a list of the areas that will not be specifically addressed. We haven't tested this functionality because we didn't have enough time in our work to test it. Furthermore, it is impossible to predict every possible way that users may interact with the software. This means that it may not be possible to test every feature in every possible scenario. For example:

Customer can see their profile: A customer can see his/her detailed profile. This feature was not tested in this software.

Customer can change profile picture: A customer can change his/her profile picture which was not tested.

Customer can navigate between food menu: Customer can navigate between every food menu which was not tested.

Customer can confirm order: A customer can confirm his/her food order after the selection from the food menu.

6. TESTING APPROACH

6.1 Testing Levels

The testing for the VhojonBilash is consisted of Unit, Integration, System and Acceptance testing levels. It is hoped that at least one full time independent test person will be available for system and integration testing. However, due to the budget constraints and deadline of the project, most of the testing will be done by the test manager with the development teams' participation.

I. **Unit Testing:** Firstly, we have done unit testing. Here we have tested individual software units or components. We have to ensure that every module works properly. Mainly, when a developer writes codes, then the programmer does this kind of testing. Thus, white box testing is used in this unit.

II. **Integration testing:** After finishing the unit testing, we have done the integration testing. As we know that we have to merge every small unit and module after constructing them. For this reason, we have to ensure that all the software modules are integrated correctly and tested as a group. Bottom-up Integration technique is used for integration testing and the main goal is to find any kind of flaws after integrating each modules.

III. **System Testing:** When we combine every module, include every feature, build a complete system, and then test it, we are performing system testing. Verification of the software requirements is done in this part. Every feature and requirement was examined.

Black box testing is carried out at this level same as white box testing is carried out when the project is in a modular state.

IV. **Acceptance Testing:** This kind of testing is done at the ending stage of a software. End users are the main tester here. Acceptance testing, a testing technique performed to determine whether or not the software system has met the requirement specifications.

6.2 Test Tools

- To test the entire program, we have used SELENIUM.

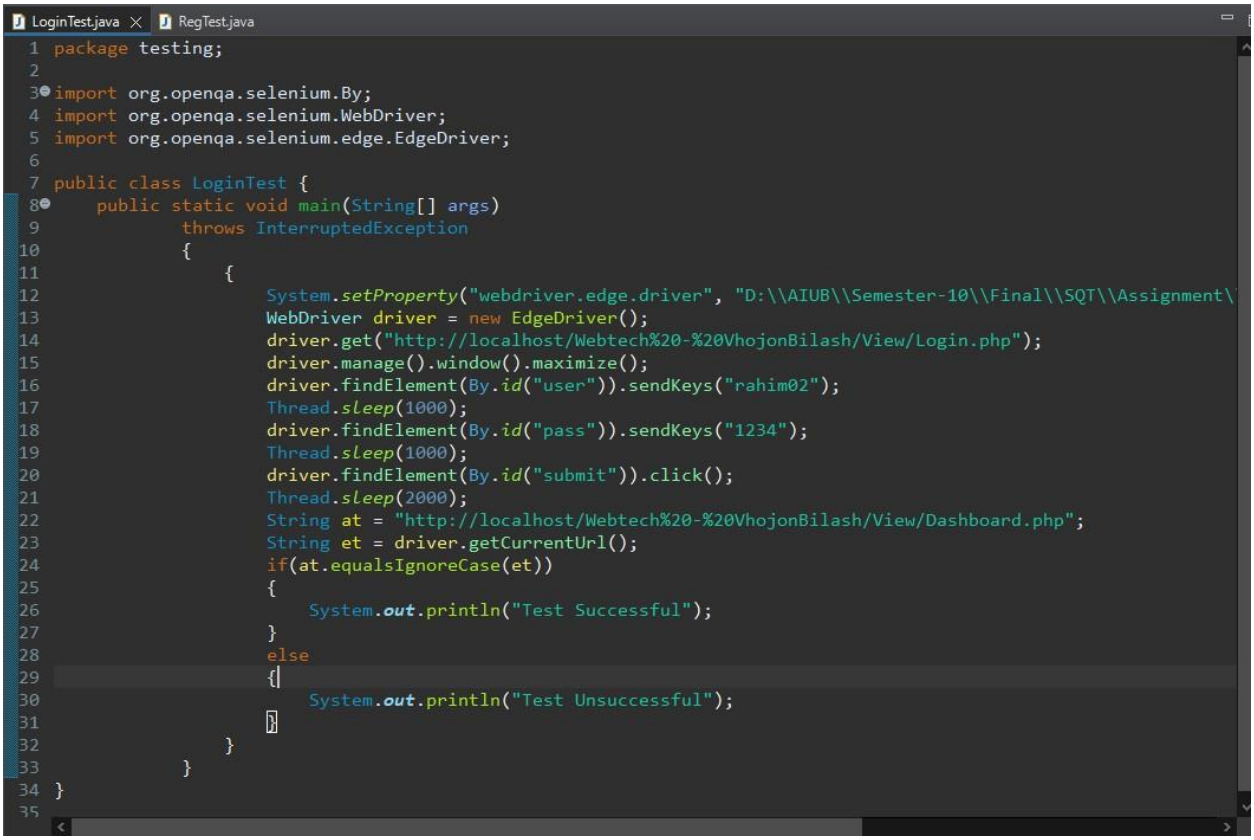
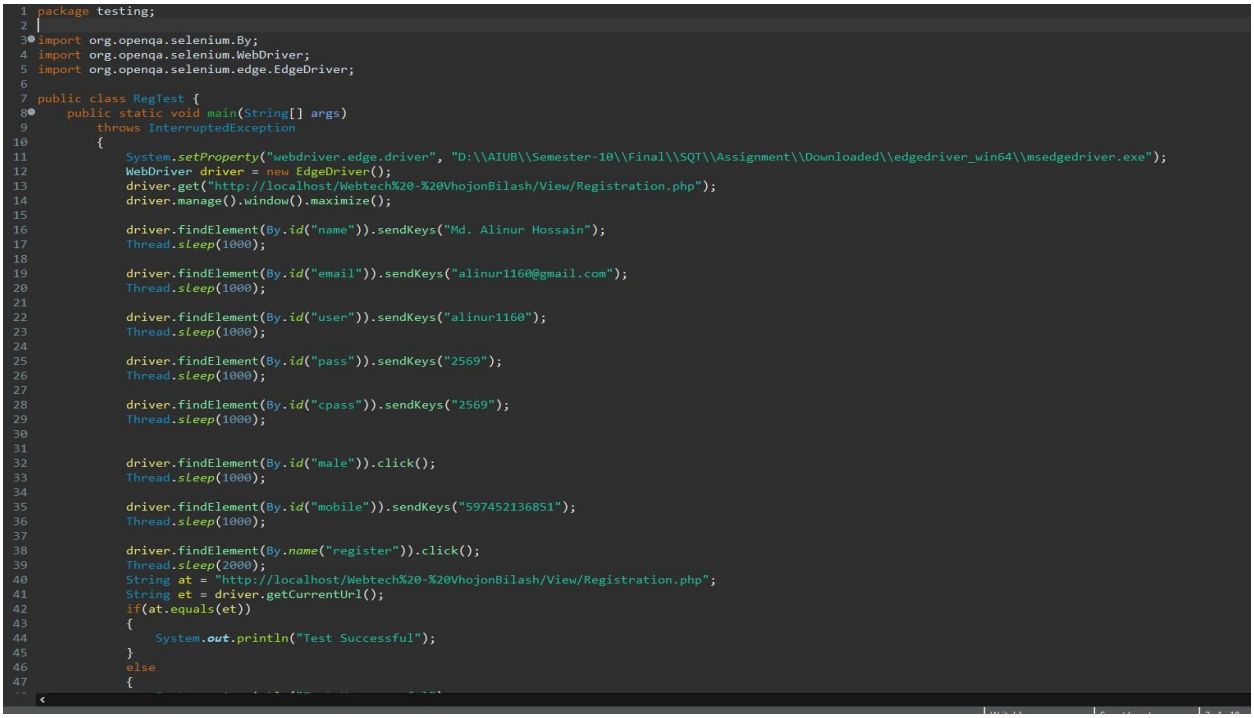


Fig:1 – Testing Login page using selenium



```
package testing;

import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.edge.EdgeDriver;

public class RegTest {
    public static void main(String[] args)
        throws InterruptedException
        {
            System.setProperty("webdriver.edge.driver", "D:\\AIUB\\Semester-10\\Final\\SQT\\Assignment\\Downloaded\\edgedriver_win64\\msedgedriver.exe");
            WebDriver driver = new EdgeDriver();
            driver.get("http://localhost/Webtech%20-%20VhojonBilash/View/Registration.php");
            driver.manage().window().maximize();

            driver.findElement(By.id("name")).sendKeys("Md. Alinur Hossain");
            Thread.sleep(1000);

            driver.findElement(By.id("email")).sendKeys("alinur1160@gmail.com");
            Thread.sleep(1000);

            driver.findElement(By.id("user")).sendKeys("alinur1160");
            Thread.sleep(1000);

            driver.findElement(By.id("pass")).sendKeys("2569");
            Thread.sleep(1000);

            driver.findElement(By.id("cpass")).sendKeys("2569");
            Thread.sleep(1000);


            driver.findElement(By.id("male")).click();
            Thread.sleep(1000);

            driver.findElement(By.id("mobile")).sendKeys("597452136851");
            Thread.sleep(1000);

            driver.findElement(By.name("register")).click();
            Thread.sleep(2000);
            String at = "http://localhost/Webtech%20-%20VhojonBilash/View/Registration.php";
            String et = driver.getCurrentUrl();
            if(at.equals(et))
            {
                System.out.println("Test Successful");
            }
            else
            {
```

Fig:2 – Testing Registration page using selenium

6.3 Meetings

Regular meetings between developers and testers play a vital role in creating a successful software product. Through effective collaboration and communication, developers and testers can discuss their roles, share ideas, clarify requirements, and identify potential issues early on in the development process. This leads to the early detection of bugs, improved software quality, a better understanding of user needs, and greater efficiency in the development cycle. Ultimately, regular meetings between developers and testers foster a collaborative working environment that helps to ensure a successful software product. For this reason, the test team will meet once or twice every week to evaluate the progress of each other. If there is any emergency, then extra meetings can also be arranged.

7. TEST CASES/TEST ITEMS

7.1 Registration:

Project Name: VojonBilash			**Test Designed by:** Faiza Tasnim	
Test Case ID: Registration_1			**Test Designed date:** 08/04/23	
Test Priority (Low, Medium, High): High			**Test Executed by:** Md. Alinur Hossain	
Module Name: Sign up			**Test Execution date:**09/04/23	
Test Title: Sign up with valid data.				
Description: Test the website for registration name, username, email, password, gender and mobile number.				
Precondition (If any):				
Test Steps	**Test Data**	**Expecte dResults**	**Actual Results**	**Status (Pass/Fail)**
1. Go to the website 2. Enter name 3. Enter email 4. Enter username 5. Enter password 6. Retype password 7. Choose gender 8. Enter mobile number 9. Click submit	Username: alinur Gmail: alinur@gmail.com Password:12345 Confirm password: 12345	Users will be signed up to the website	As expected,	Pass
Post Condition: User is validated with database and account details are stored in the database.				

7.2 Log In:

Project Name: VojonBilash			**Test Designed by:** Faiza Tasnim	
Test Case ID: login_2			**Test Designed date:** 08/04/23	
Test Priority (Low, Medium, High): High			**Test Executed by:** Md. Alinur Hossain	
Module Name: Login			**Test Execution date:**09/04/23	
Test Title: Verify login with valid data				
Description: Test website login page valid username and password.				
Precondition (If any):				
Test Steps	**Test Data**	**Expected Results**	**Actual Results**	**Status (Pass/Fail)**
1. Go to the website 2. Enter email id 3. Enter password 4. Click submit	Username: alinur@gmail.com Password: 4567	User should login into the application	As expected,	Pass
Post Condition: User is validated with database and successfully login to account. The accountsession details are logged in the database.				

7.3 Logout:

Project Name: VojonBilash			**Test Designed by:** Faiza Tasnim	
Test Case ID: Logout_3			**Test Designed date:** 08/04/23	
Test Priority (Low, Medium, High): Medium			**Test Executed by**: Md. Alinur Hossain	
Module Name: Logout			**Test Execution date:** 09/04/23	
Test Title: Logout				
Description: Test the website Logout option				
Precondition (If any):				
Test Steps	**Test Data**	**Expected Result**	**Actual Result**	**Status (Pass/Fail)**
1. Go to the site 2. Sign in to the website 3. Click the "Logout"button	No data needed.	User will besigned out from the system	As expected,	Pass
Post Condition: User goes back to the home page				

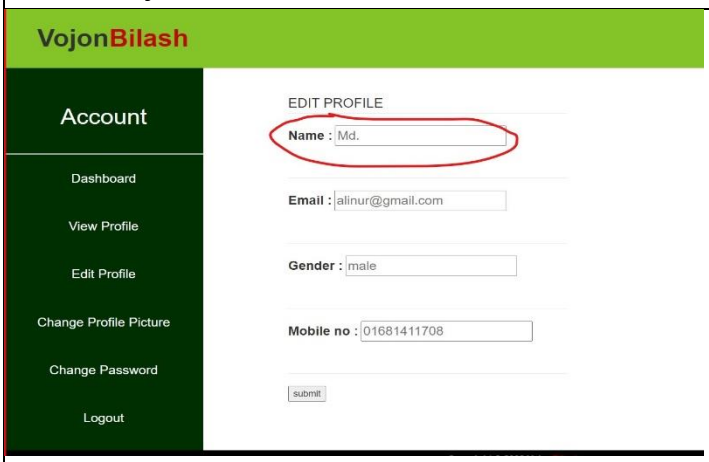
7.4 Edit profile:

Project Name: VojonBilash	**Test Designed by:** Faiza Tasnim
Test Case ID: EditProfile_4	**Test Designed date:** 08/04/23
Test Priority (Low, Medium, High): High	**Test Executed by**: Md. Alinur Hossain
Module Name: Account	**Test Execution date:** 10/04/23
Test Title: Test the edit profile feature	
Description: Change user Name, Email, Gender and Mobile number. if profile gets updated or not.	

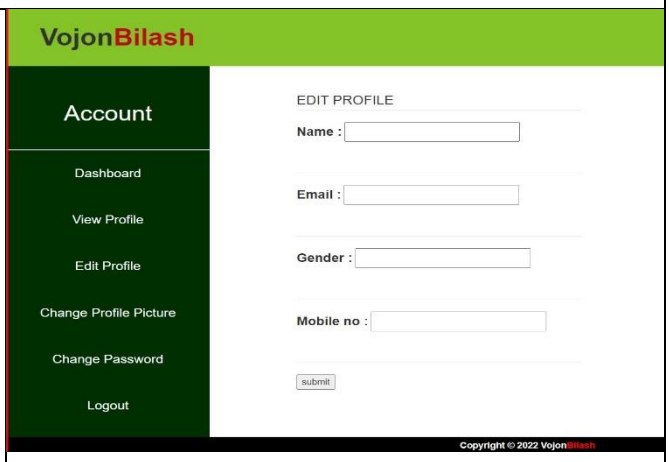
Precondition (If any): User must have his/her account.

Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Go to the site 2. log in to the software. 3. Go to the edit profile 4. Update the username or password	Username: alinur@gmail.com Password: 4567	Profile updated	Profile not updated	Fail

Post Condition: The updated Name, Email, Gender or Mobile Number has been stored in the database successfully.



Placeholder tag was used to show the details in the text input area but somehow it was not working properly as it can be seen in the screenshot.



Most likely the Placeholder tag was not supporting the spaces between words and to avoid this bug we have removed the tag.

7.5 Change password:

Project Name: VojonBilash	**Test Designed by:** Sadia Afrin Sara
Test Case ID: ChangePassword_5	**Test Designed date:** 08/04/23
Test Priority (Low, Medium, High): High	**Test Executed by**: Md. Alinur Hossain
Module Name: Account	**Test Execution date:** 09/04/23
Test Title: Test the change password	
Description: Password is updated or not.	
Precondition (If any): User must have his/her account.	

Test Steps	**Test Data**	**Expected Results**	**Actual Results**	**Status (Pass/Fail)**
1. go to the site 2. log in to the software. 3. Go to the edit profile 4. Click the “change password” button	Old Password=4567 New Password=1234 Retype Password=1234	Password change successful	Password Updates Successfully	Pass

Post Condition: The updated password is stored in the database

7.6 Food Searching:

Project Name: VojonBilash	**Test Designed by:** Sadia Afrin Sara
Test Case ID: FoodSearching_6	**Test Designed date:** 08/04/23
Test Priority (Low, Medium, High): High	**Test Executed by**: Md. Alinur Hossain
Module Name: Food Menu	**Test Execution date:** 10/04/23
Test Title: Food Search	
Description: Test the working criteria of search food	

Precondition (If any): Entire name of the food is not necessary. Some beginning words of the food name will show food suggestions.

Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Go to the Restaurant 2. Click on search food suggestions 3. Enter food name	Search: fry or fr to see the results.	User should get the suggestions	As expected,	Pass

Post Condition: User got the suggestions as he/she wanted.

7.7 Food Order:

Project Name: VhojonBilash				**Test Designed by:** Sadia Afrin Sara
Test Case ID: Food_Order_7				**Test Designed date:**8/4/2023
Test Priority (Low, Medium, High): High				**Test Executed by:** Md Alinur Hossain
Module Name: Food Menu				**Test Execution date:**10/4/2023
Test Title: Food order				
Description: Test website Food Order page				
Precondition (If any): User must add food into the cart or order list				
Test Steps	**Test Data**	**Expected Results**	**Actual Results**	**Status (Pass/Fail)**
1. Go to the Food Order. 2. Click add button next to a food.	Click Add	User should be able to add a food in the order list	As expected,	Pass
Post Condition: User has successfully added food in the order list.				

7.8 Delete Order:

Project Name: VhojonBilash				**Test Designed by:** Sadia Afrin Sara
Test Case ID: Delete_Order_1				**Test Designed date:**7/4/2023
Test Priority (Low, Medium, High): High				**Test Executed by:** Md Alinur Hossain
Module Name: Food Cart				**Test Execution date**:9/4/2023
Test Title: Delete order				
Description: Test the delete order				
Precondition (If any):				
Test Steps	**Test Data**	**Expected Results**	**Actual Results**	**Status (Pass/Fail)**
1. Go to the Food Order 2. Click Delete Order 3. Click Delete button next to a food.	Click Delete	User should be able to delete a food in the order list	As expected,	Pass
Post Condition: User has successfully deleted food item from the food order list.				

8. ITEM PASS/FAIL CRITERIA

We have conducted 8 test cases to find defects from the program. At first, when applying the test case to the system, 95% of the test cases were passed successfully and 5% test were failed. The test cases failed due to session related issue. However, after conducting the test cases, we have successfully solved the problem.

9. TEST DELIVERABLES

- **Acceptance test plan:** The users who were conducting the acceptability tests has become successful. The program was easy to use and the interface were clean.

- **System/Integration test plan:** System integration was completed successfully. All other modules, functionalities including the database were also working properly.

- **Unit test plans/turnover documentation:** The unit testing was done successfully by the developers.

- **Screen prototypes:** Several prototypes were made and done the testing in the initial stage of the design process.

- **Report mock-ups:** Though, this is a project report, for this reason we have not created any mock-up reports.

- **Defect/Incident reports and summaries:** We have found some defects in the software but those have been fixed while conducting tests.

- **Test logs and turnover reports:** All of the tests were completed successfully. Bugs or defects were removed.

10. STAFFING AND TRAINING NEEDS

We have total 5 staffs for our test plan

- 1 Project manager
- 1 Quality Assurance Engineer
- 3 Developers

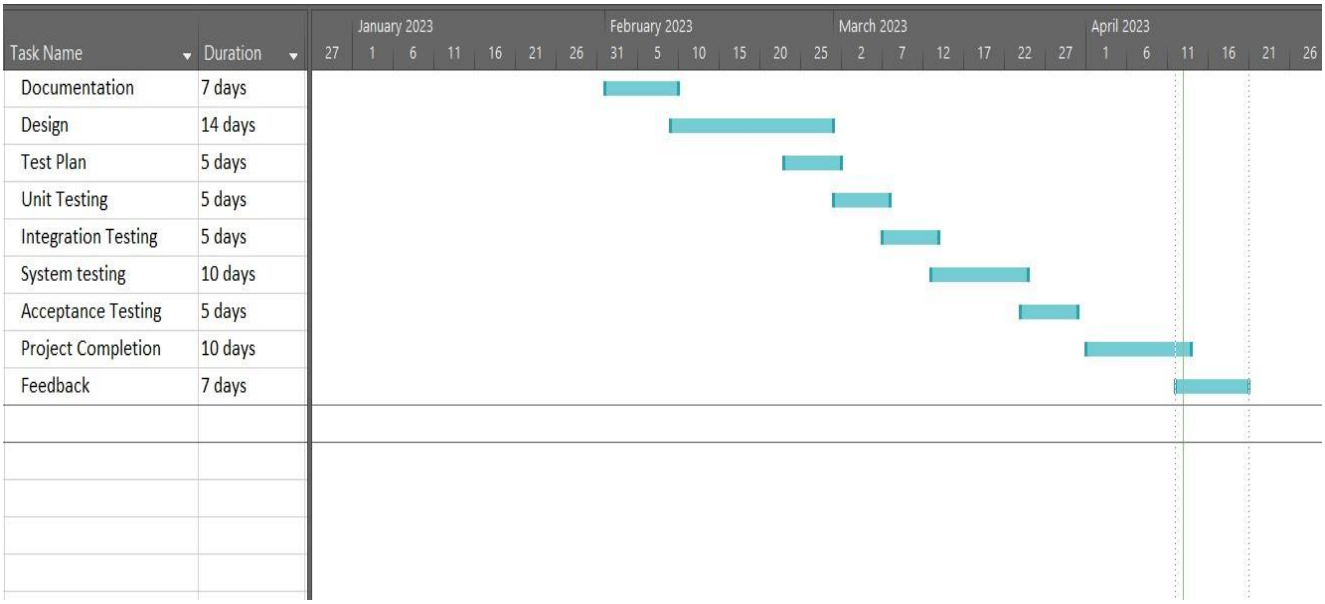
The developers and tester(s) will need to be trained on the basic operations of the EDI interface. This will include understanding the communication protocols, data formats, error handling, and other technical aspects of the interface. It is important that the team is fully trained on these topics in order to ensure the smooth integration of the solar tracking system with other systems in the energy grid. The Developers and Project manager must be familiar with how to use the test software. Furthermore, online dependency is not enough but also the knowledge about the testing should be applied. Developers need more effort to develop a software.

11. Responsibilities

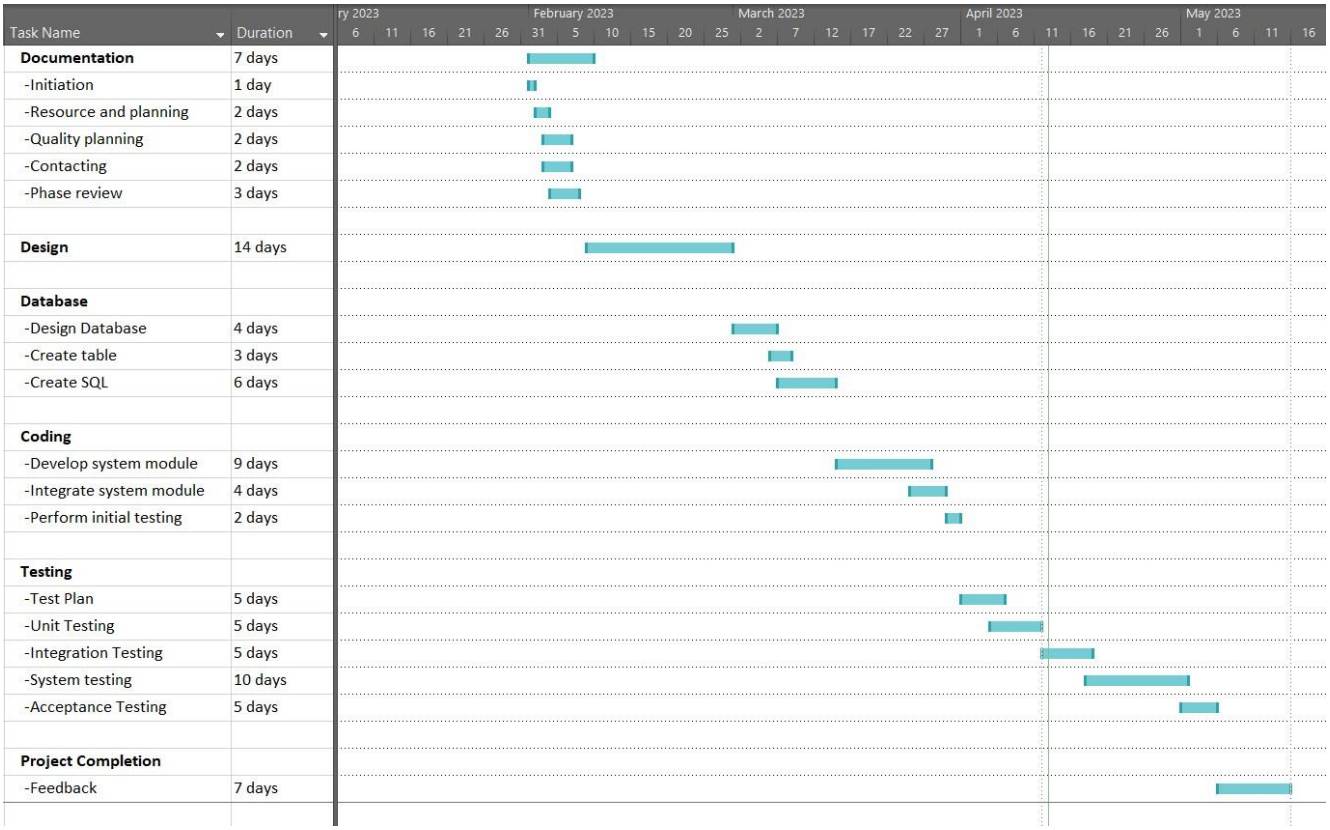
Serial	Name	Role	Responsibilities
1	Md. Alinur Hossain	Project manager	1. Acceptance test Documentation and Execution. 2. System and Detail Design Reviews. 3. Test procedure and rules. 4. Change control and regression testing.
2	Faiza Tasnim	Testing manager	1. Every Test Documentation and Execution. 2. System and Detail Design Reviews. 3. Test procedure and rules. 4. Change control and regression testing.
3	Sharia Tasnim Adrita, Sadia Afrin Sara,	Developer team	1. System/Integration and Unit test Documentation and Execution. 2. System and Detail Reviews. 3. Screen and Report prototype reviews. 4. Change control and Regression testing.
4	Sharia Tasnim Adrita, Sadia Afrin Sara, Faiza Tasnim,	Testing team	1. Every Test Documentation and Execution. 2. System and Detail Design Reviews. 3. Test procedure and rules. 4. Change control and regression testing. 5. Screen and Report prototype reviews.

12. TESTING SCHEDULE

Project Plan:



Project Schedule:



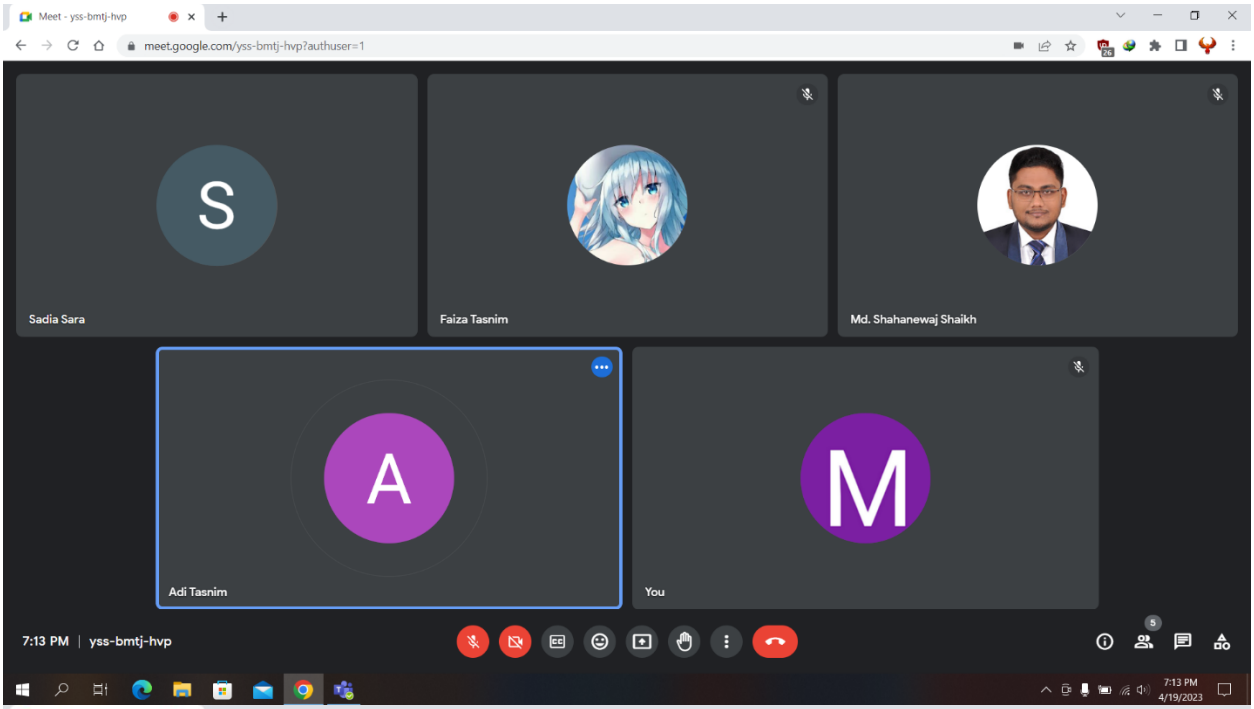
13. PLANNING RISKS AND CONTINGENCIES

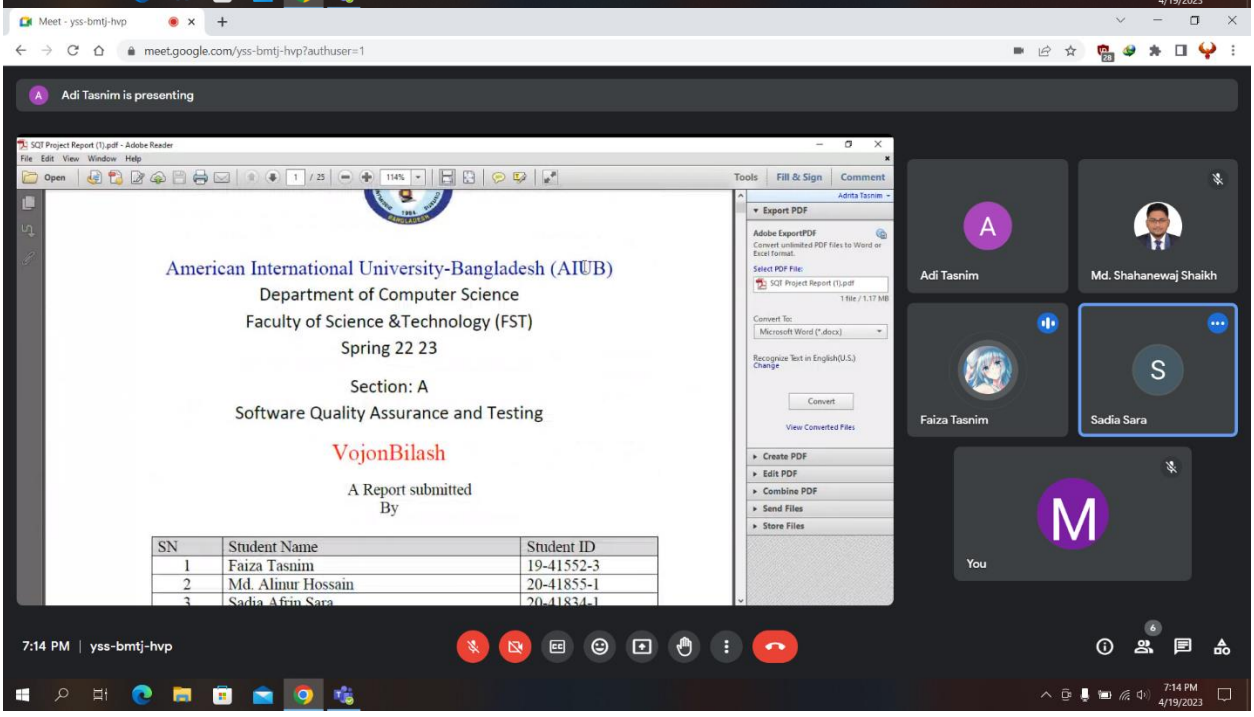
Risk	Probability	Impact	Mitigation
Error in database	Low	Low	Testing the database related features frequently and maintain daily update.
Error in function	Low	Low	Testing the website frequently and maintain daily update.
Error while giving invalid input	Medium	Medium	The users or testers have to test the input fields regularly by giving correct input data.
Loss of Data	High	High	Maintain security checkup regularly.

14. APROVALS

Project Sponsor	Pass
Development Management	Pass
EDI Project Manager	Pass
RS Test Manager	Pass
RS Development	Pass
Reassgined Sales	Pass
Order Entry EDI team Manager	Pass

Meeting:


Meet - yss-bmtj-hvp
meet.google.com/yss-bmtj-hvp?authuser=1
S
Sadia Sara
Faiza Tasnim
Md. Shahanewaj Shaikh
A
Adi Tasnim
M
You
7:13 PM | yss-bmtj-hvp
7:13 PM
4/19/2023


Meet - yss-bmtj-hvp
meet.google.com/yss-bmtj-hvp?authuser=1
Adi Tasnim is presenting
American International University-Bangladesh (AIUB)
Department of Computer Science
Faculty of Science &Technology (FST)
Spring 22 23
Section: A
Software Quality Assurance and Testing
VojonBilash
A Report submitted
By
SN
Student Name
Student ID
1
Faiza Tasnim
19-41552-3
2
Md. Alinur Hossain
20-41855-1
Adi Tasnim
Md. Shahanewaj Shaikh
Faiza Tasnim
Sadia Sara
You
7:14 PM | yss-bmtj-hvp
7:14 PM
4/19/2023