American International University-Bangladesh (AIUB)
Department of Computer Science
Faculty of Science &Technology (FST)
Summer 21 22

Section: A
Software Quality Assurance and Testing

SMART CAFETERIA

A Report submitted
By

SN	Student Name	Student ID
1	Mohammed Mofizur Rahman	19-40120-1
2	Md. Mridul Hosen	18-38295-2
3	Md. Abir Hossain	19-40907-2

Under the supervision of

Abhijit Bhowmik
Associate Professor & Special Assistant [OSA]
Department of Computer Science
Faculty of Science & Technology
American International University-Bangladesh

Software Test Plan

for

Smart Cafeteria

Version 1.0 approved

Prepared by,

Mohammed Mofizur Rahman
Md. Mridul Hosen
Md. Abir Hossain

American International University-Bangladesh (AIUB)
16 August, 2022

Checked By Industry Personnel

Name:

Designation:

Company:

Sign:

Date:

Table of Contents

Revision History

Revision	Date	Updated by	Update Comments
0.1	2022.07.04	Md. Mridul Hosen	First Draft
0.2	2022.07.09	Mohammed Mofizur Rahman	Second Draft
0.3	2022.07.04	Md. Abir Hossain	Third Draft
0.4	2022.07.09	Mohammed Mofizur Rahman	Forth Draft
0.5	2022.07.04	Md. Mridul Hosen	Fifth Draft
0.6	2022.07.09	Mohammed Mofizur Rahman	Sixth Draft
0.7	2022.07.04	Md. Mridul Hosen	Seventh Draft
0.8	2022.07.09	Md. Abir Hossain	Eighth Draft

1. TEST PLAN IDENTIFIER: TP_SmartCafeteria_1.02

2. REFERENCES

a) Software Quality And Testing Course PowerPoint Slides
b) What is Postman Automation? - GeeksforGeeks
c) https://sourceforge.net/projects/projectlibre/
d) https://www.postman.com/automated-testing/
e) https://www.w3schools.com/
f) Kiosk System Described
g) Selenium WebDriver Documentations

3. INTRODUCTION

3.1 Background to the Problem

We have a cafeteria in our AIUB campus at Annex 1. But sometimes the cafeteria is filled up with bunch of students. Since the cafeteria is filled with and it can be seen from the outside through glass, it is quite inconvenient for students and faculties to order food. Moreover, some of the students feel uneasy to go inside and order their desired items.

The main problem of the cafeteria is that, the cafeteria size is small compared to the students and the faculties seating capacity. Though it is the time of the modern world, in our cafeteria we have no embedded or smart system through which we can order food from the outside of the cafeteria. If we can solve the problem of having a smart device, the students and faculties can have their lunch or snacks or their desired food easily.

3.2 Solution to the Problem

The main objective of our project is to utilize the modern technology in terms of using the minimum criteria and the space of our university. Since some of the students and faculties do not get enough time to sit and order in their rush hour, we need a solution that will help them out in a very convenient and easy way.

It is hoped that our system will be an appropriate solution as it is unique and easy to use and also time consuming. A student or faculty can see the availability of food items and can also order directly from the Kiosk system of the cafeteria of AIUB. The order list will be added to cart and the user can pay by cash or using Bkash. After approving the order, the user can take the order from the open window of the cafeteria. So, the system would save time and the space of the cafeteria.

4. REQUEIREMNT SPECIFICATION

4.1 System Features

1. Student/User features:

A Student/User can:

- **1.1** Explore food items
- **1.2** Check the availability of food items
- **1.3** Order food items
- **1.4** Add to Cart
- **1.5** Confirm the order through the cart items
- **1.6** Cancel Cart items or Order list
- **1.7** Payment with cash (Scanning Machine) or Bkash
- **1.8** Give ratings to the food items
- **1.9** Give ratings to the employees

2. Employee Features:

An Employee can:

- **2.1** Check order List
- **2.2** Update availability of food items
- **2.3** Confirm order by Bkash or Cash
- **2.4** Delivery the order
- **2.5** Show the waiting time to the customer side Kiosk system
- **2.6** Complain about the users.

3. Manager Features:

A Manager can:

- **3.1** Add new food items
- **3.2** Update food items
- **3.3** Delete food items
- **3.4** Update the price of the food items
- **3.5** Check accounts
- **3.6** Give special offers
- **3.7** Control Employee data

4.2 System Quality Attributes

- **Usability:** Since usability can be measured in terms of ease of use, the application should be user-friendly. A regular user should be able to order his/her desired food item in average of 3 minutes and a maximum of 5 minutes time period. As the system features are made easy to learn as navigation is very simple. It's also Easy for a new or infrequent users to learn to use the system.

- **Maintainability:** Maintainability is the term that defines that how easily the maintenance team can perform their task. The main task of our maintenance team is to solve the bugs, adding something new or changing some features. One of our maintenance programmers shall be able to modify any features with 20 labor hours or less of development effort.

- **Efficiency:** Since efficiency is a major system quality attribute, it is measured in terms of time required to complete any task given to the system. The application has passed with a large scale in efficiency as it is easy to use and also concise.

- **Modularity**: Every system that is developed module wise. In this kiosk system there are many modules added and those are integrated to make the whole system, so that bugs can be easily detected in any module and then the defects have to be solved that particular module. We don't have to think about other modules so it makes the work easy for our tester. In addition of that we can also add new features in our system as it has been developed module wise.

- **Reliability And Correctness:** In software development process there **is** no software is free from defects or minor bugs. Our software gives the exact output what user wants and also the correctness of the kiosk system has been ensured. Our software does not crash randomly as it has been tested and no false output is generated so it's more reliable to use for our users.

- **Testability:** Testability refers to a term that is measured on how easily the testing team can perform their task. Testability is very much depended on modularity. Since our system is developed module wise so test engineers will have to go through an easy-going process. They don't have to check every module to fix bugs.

- **Flexibility:** The effort required to modify an operational program is called flexibility. A maintenance programmer who has at least one year of experience supporting this product shall be able to make a new copy output available to the product, including code modifications and testing, with no more than 2 hours of labor work.

4.3 System Interface

Here, system interfaces like Welcome Page showing all the food items, Order List from the User view with payment options such as Bkash and Cash, Employee view Orders and Database for updating food lists for Manager are described below:

1. Welcome Page for Students/Users

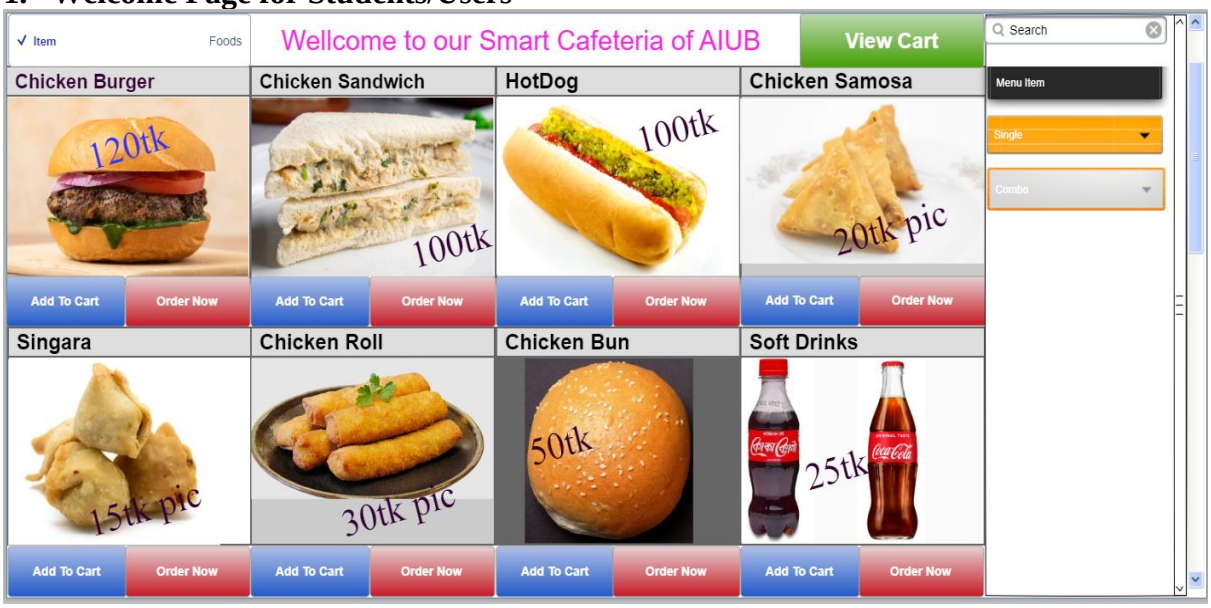

Fig.01: Home page welcoming Students/Users.
This home page shows all the food that is available to the cafeteria. A student can see item price, select multiple items, see the View Cart and many more. A student can cancel his order any time before the payment.

2. Add to cart with payment options

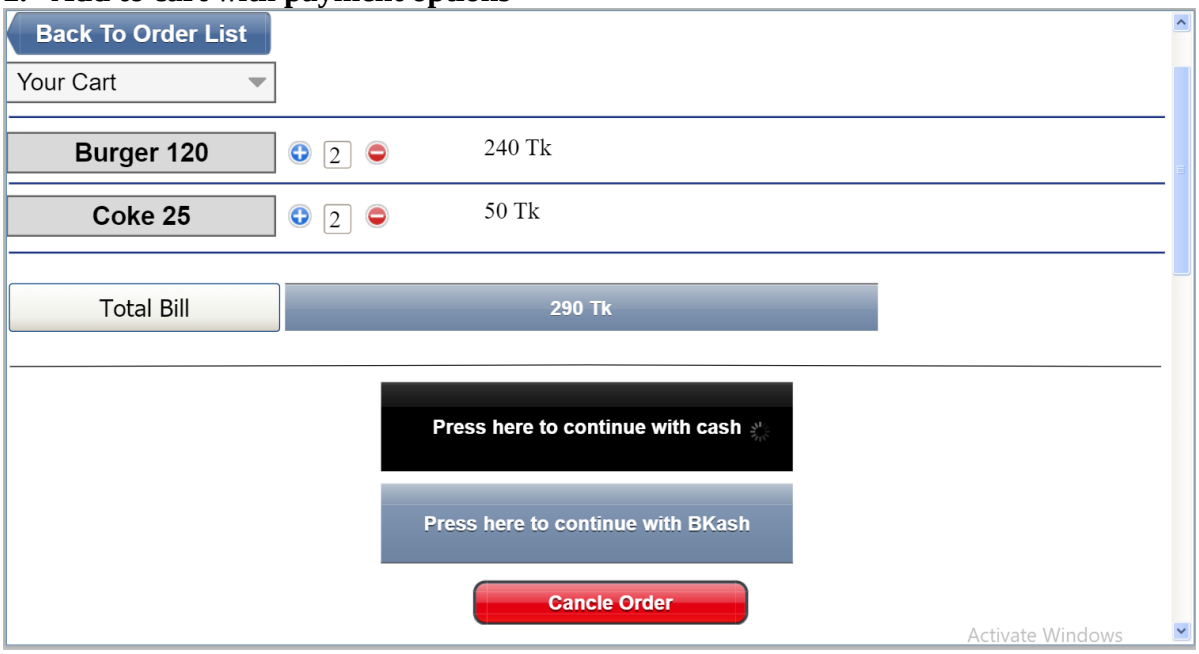

Fig.02: This is the cart that shows the items that a student added to his/her cart selecting one or multiple items. This page has two payment options. A student can pay using Bkash or with cash.

3. Order Interface for Employee

Fig.03: This is the interface for employee to see the orders that the students ordered using the Kiosk system. An employee can set waiting time as required for the students who ordered.

4. Database for Manager to Update Food Items

Fig.04: The database is for the manager from where manager can add new food or drinks, and delete items.

4.4 Project Requirements

Effort Estimation:

Our project is to develop an application named “Smart Cafeteria of AIUB”.

Development Time = 6 Weeks

Required number of peoples = 4

Budget Estimation:

4 developers and/or engineers working of 2 months:

Duration in weeks = 6 weeks

Office days = 5 days

Working hours = 8 Hours

So, per week working hours is = (5*8) hours

= 40 hours

Hence, Total Working hours is = (40*6) hours

= 240 hours.

Developer salary is = 1000 BDT

Total developers Salary = (1000*240) BDT

= 240000 BDT

Basic Resources or components of Kiosk System building process are:

- Touch Screen Monitor,
- Internal Computer,
- Kiosk Enclosure

Standard features added:

- External Keyboard,
- Card Readers,
- Barcode Readers,
- Printers

Expanse	Amount	Total Amount
Salary for 4 developers		240,000 BDT
2 months office rent	2*20,000	40,000 BDT
Electricity and other Utility Costs		12,000 BDT
6 weeks Maintenance cost	6*10,000	60,000 BDT
Total Cost		352,000 BDT
15% of total cost(profit)		52,800 BDT
Total budget:		**404,800 BDT**

5. FEATURES NOT TO BE TESTED

The following is a list of the areas that will not be specifically addressed. All testing in these areas will be indirect as a result of other testing efforts. For example:

i. A student/User can contact manager. In our Kiosk application there is a feature where a user can contact manager through messaging. This feature is not to be tested in this release of the software.

ii. A student can login with their university id into the Kiosk system. The feature is on continuous process but it is not included in this release of the software.

iii. An employee can complain about the student mentioning the university id of the students. The feature is yet to be tested.

iv. A user can add vouchers during the payment. It will be released but not tested as a functional part of the release of this version of the software.

6. TESTING APPROACH

- **Unit Testing:** First we'll do the Unit Testing. during this testing we'll tests individual software units or components. as an example, each class or method. The goal is to verify that each unit of software code works as intended. This testing is applied by developers throughout the event (coding) phase of an application. during this step, we are visiting implement “White Box Testing” technique.

- **Integration Testing:** then within the second part we are visiting do the blending. during this testing we are visiting confirm that everyone the software modules are integrated logically and tested as a bunch. Our project is formed of several software modules written by Four programmers. The goal of this level of testing is to hunt out flaws within the way various software modules interact when they're integrated. during this step, we are visiting follow the “Bottom-up Integration” technique.

- **System Testing:** Then we are visiting do the system testing. Through the system testing we'll test of full-featured, fully integrated system. Then we'll verify if it meets all the necessity. Black-box testing falls under this condition. So, at this level, we are visiting follow “Black Box Testing” technique.

- **Acceptance Testing:** The last phase of our testing is Acceptance Testing. This test is visiting be done to check whether any defect missed during the functional testing phase. At this level, we are visiting follow the “Black Box Testing” technique. After that, we may run unit tests again.

6.1 **Testing Levels**

- SYSTEM/INTEGRATION Testing is performed by the test manager and development team leader with assistance from the individual developers PRN. Programs will enter into System/Integration test finally critical defects are corrected. there's a bit around for the error.

- UNIT TESTING is done by the developer, then the results are visiting be authorized by the team lead. Before unit testing is additionally accepted and passed on to the test person, the programmer must submit proof of unit testing (test case list, sample output, data printouts, defect information). The test person will receive all unit test information moreover.

- The customer will do ACCEPTANCE testing with the assistance of the test manager and development team leader. After the System/Integration test is completed, the acceptance test will run in parallel with the prevailing manual process for two weeks.

6.2 **Test Tools**

Here, for the testing purpose Postman, a solid testing tool was used to test the connection from code to the browser. Since Postman is easy to use and update things from the database, the executed results were right and also not so time consuming. Using Postman, the test results were seen quicker than the browser itself.

Postman- Testing tool

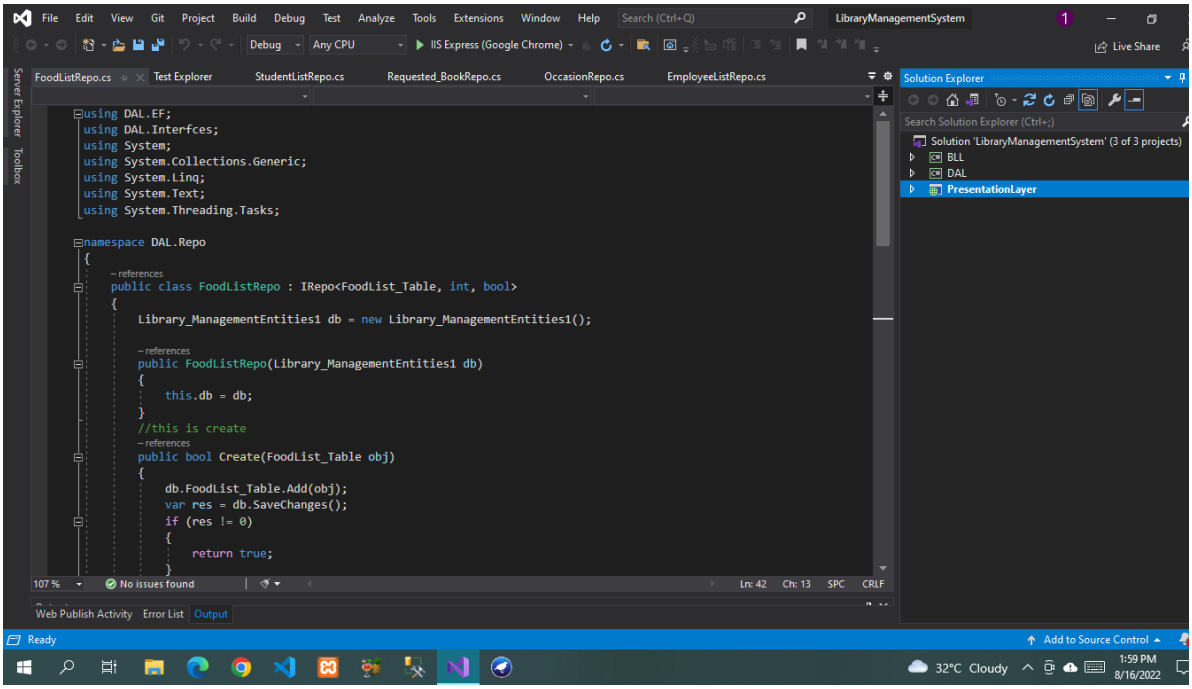

Fig.01: Smart Cafeteria Project Code on Visual Studio 2019.

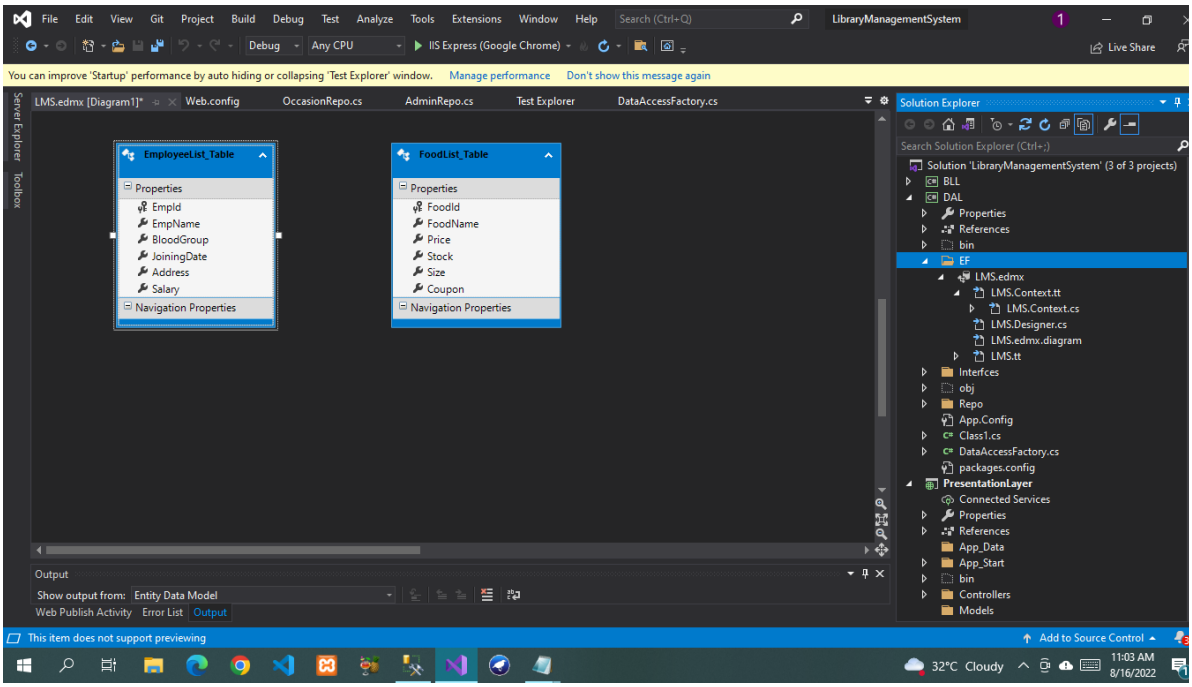

Fig.02: Smart Cafeteria Project Table on Visual Studio 2019.

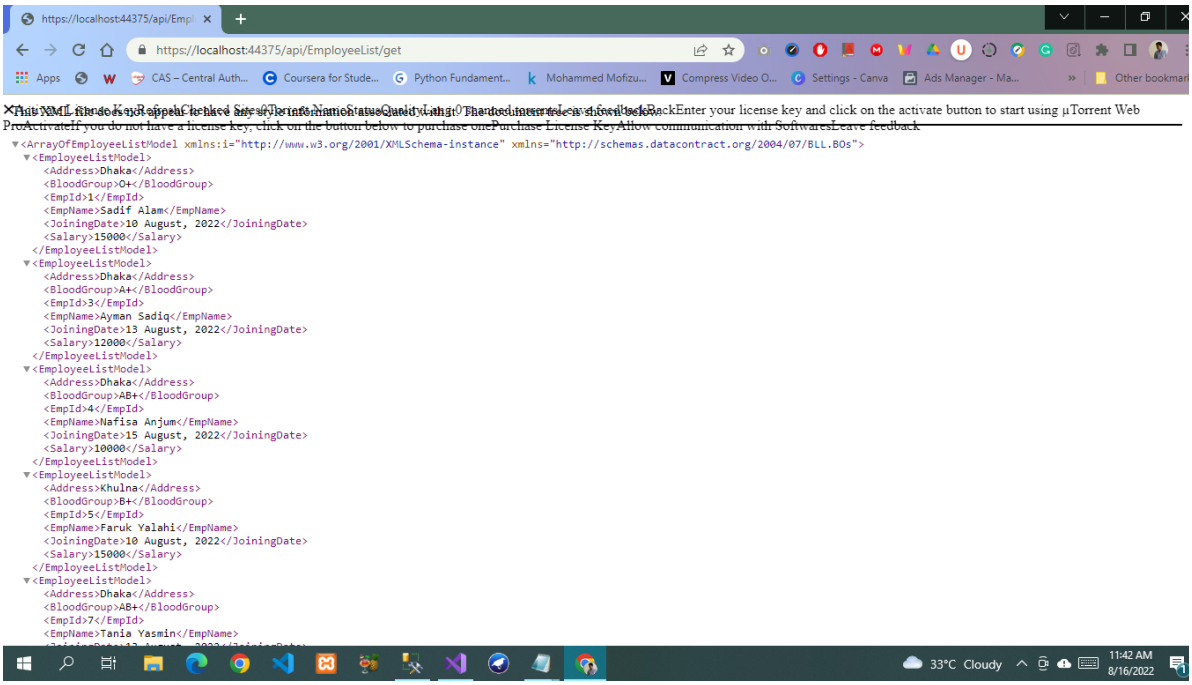

Fig.03: Smart Cafeteria Employee List [Get Method] on Browser.

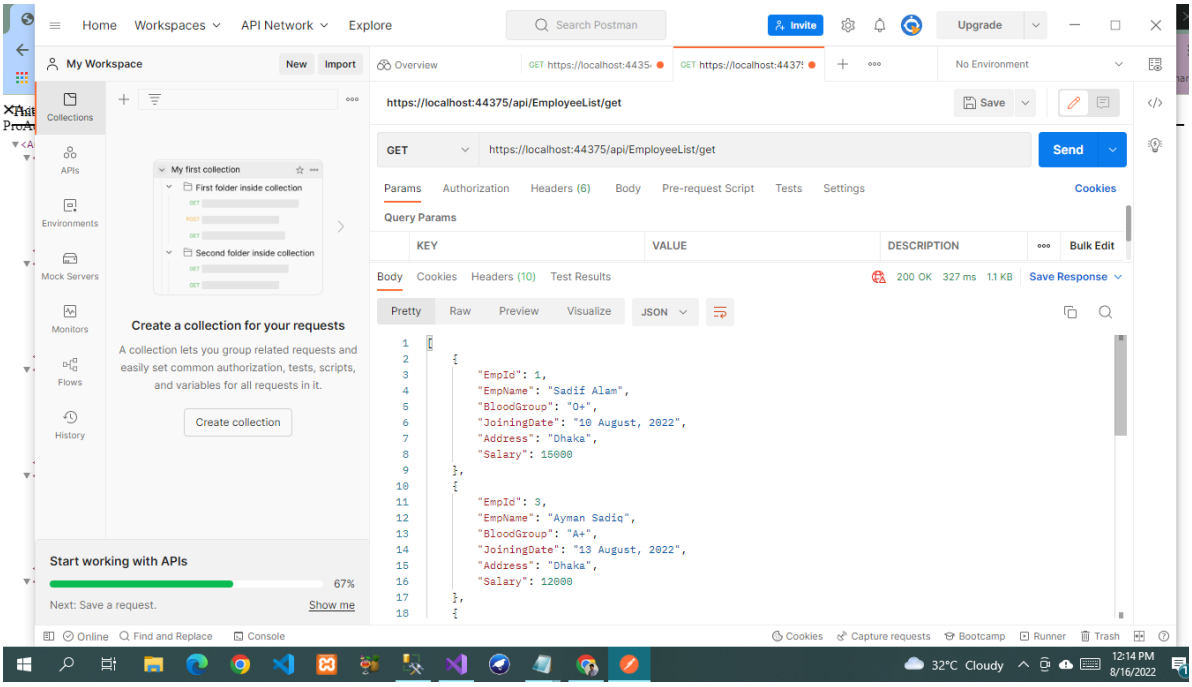

Fig.04: Employee List is successfully executed on Postman testing tool.

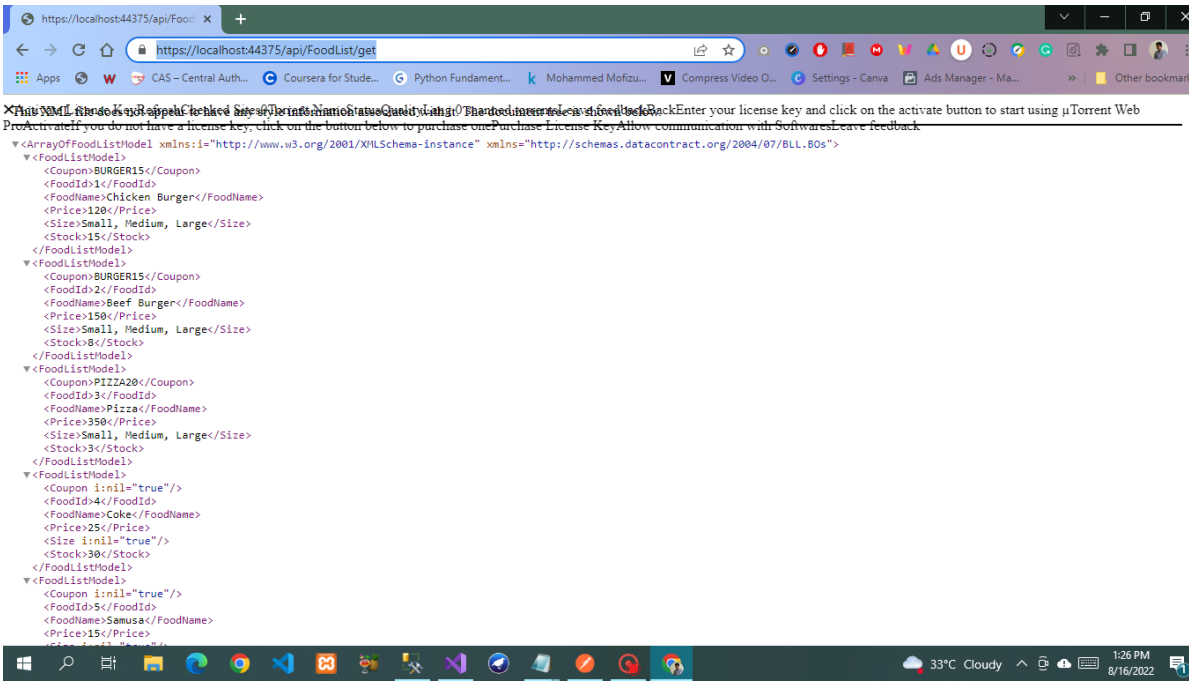

Fig.05: Smart Cafeteria Food List [Get Method] on Browser.

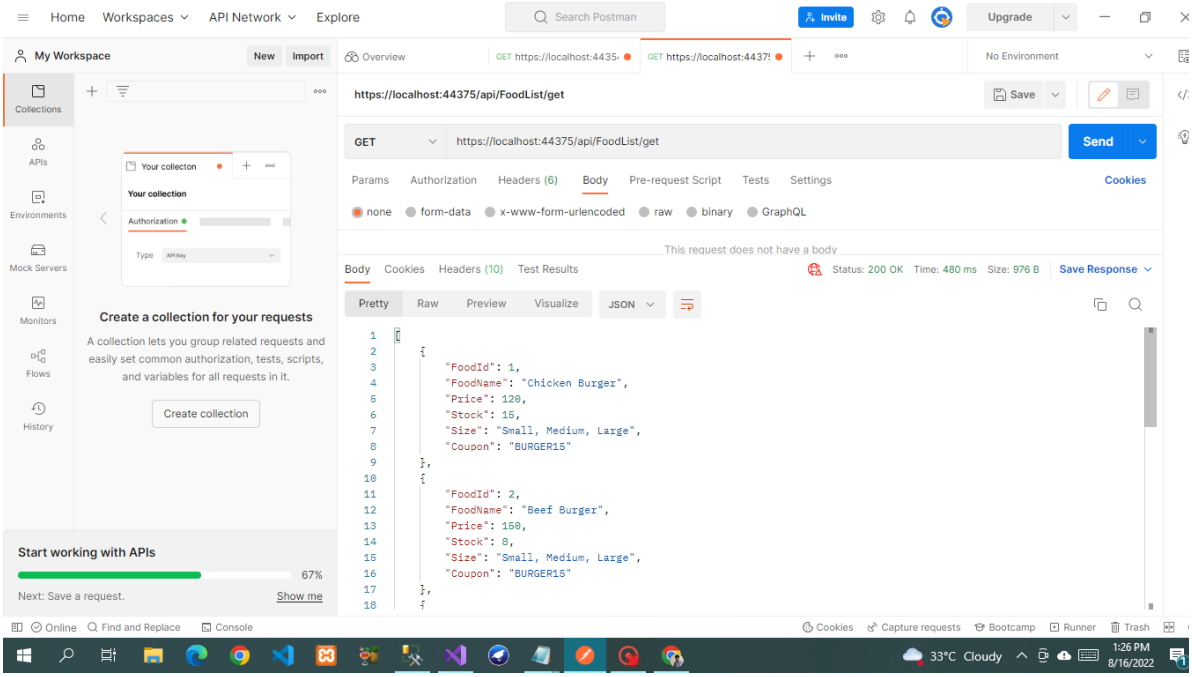

Fig.06: Food List is successfully executed on Postman testing tool.

6.3 **Meetings:**

The test team met twice in a week to judge the quality of the system and fixed a standard date to deliver the next product and prepared for the upcoming processes and to spot error trends and problems as early as possible. The test team leader met with developers and testers time to time to

ensure the progress of the integration system. On the other hand, the project manager once every period of time ensured whether the overall process is going on a flow or not so that, the final product can be delivered on time, within budget and using optimal resources. These two meetings were scheduled on different weeks. Additional meetings were called as needed for emergency situations.

7. TEST CASES/TEST ITEMS

7.1 **View Food List**

This Test case will check that the feature is showing all the product with their corresponding details or not

Project Name: Smart Cafeteria of AIUB			Test Designed by: Mohammed Mofizur Rahman	
Test Case ID: SCA_01			Test Designed date: 10 June, 2022	
Test Priority (Low, Medium, High): High			Test Executed by: Md. Mridul Hosen	
Module Name: View Food List			Test Execution date: 11 June, 2022	
Test Title: Showing all the product with their corresponding details				
Description: Test all the products are visible or not				
Precondition (If any):				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Go to the website in Kiosk system 2. Go to the product page		User should see all the products in details with the price	As expected,	Pass
Post Condition:				

7.2 **Add to Cart**

This Test case is for checking the user whether he/she can order multiple food items.

Project Name: Smart Cafeteria of AIUB	Test Designed by: Mohammed Mofizur Rahman
Test Case ID: SCA_02	Test Designed date: 13 June, 2022

Test Priority (Low, Medium, High): Medium			Test Executed by: Md. Mridul Hosen	
Module Name: Add to cart			Test Execution date: 14 June, 2022	
Test Title: Adding items in the cart				
Description: Test whether multiple items are added in the cart				
Precondition (If any): Select more than one items from the food items				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Go to the product page. 2. Select multiple food items.		User should see all the selected items on cart.	As expected,	Pass
Post Condition:				

7.3 **Cash Payment Method**

This Test case tests the payment method. There is used two types of payment method. Users can either pay with cash or using mobile banking named Bkash.

Project Name: Smart Cafeteria of AIUB			Test Designed by: Mohammed Mofizur Rahman	
Test Case ID: SCA_03			Test Designed date: 13 June, 2022	
Test Priority (Low, Medium, High): High			Test Executed by: Md. Mridul Hosen	
Module Name: Cash Payment Method			Test Execution date: 15 June, 2022	
Test Title: Showing options for paying with cash				
Description: Test cash validation				
Precondition (If any): Items should be added to the cart.				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Add 1 burger and 1 coke 2. Pay with cash.	Pay 145 BDT cash.	The system must accept cash 145 BDT	As expected,	Pass
Post Condition:				

7.4 **Payment Method using Bkash**

This Test case tests the payment method using Bkash mobile banking.

Project Name: Smart Cafeteria of AIUB			Test Designed by: Mohammed Mofizur Rahman	
Test Case ID: SCA_04			Test Designed date: 14 June, 2022	
Test Priority (Low, Medium, High): High			Test Executed by: Md. Mridul Hosen	
Module Name: Payment Method			Test Execution date: 15 June, 2022	
Test Title: Showing options for paying with cash or Bkash				
Description: Test Bkash Payment validation				
Precondition (If any): Items should be added to the cart.				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Add 1 burgers 2. Pay using Bkash.	Pay 120 BDT using Bkash	The system must accept the Bkash transaction	As expected,	Pass
Post Condition:				

7.5 **Order Food Items**

This Test case is for checking the user whether he/she can order after the payment.

Project Name: Smart Cafeteria of AIUB	Test Designed by: Mohammed Mofizur Rahman
Test Case ID: SCA_05	Test Designed date: 14 June, 2022
Test Priority (Low, Medium, High): Medium	Test Executed by: Md. Mridul Hosen
Module Name: Order food items	Test Execution date: 15 June, 2022
Test Title: Showing ordered products on the screen	
Description: Test all the ordered products are visible or not	
Precondition (If any): Check Bkash transaction or cash acceptance.	

Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Confirm the cart list 2. Make payment		User should see all the products ordered in details with the price	As expected,	Pass
Post Condition: User should have a detailed order list chart with the price mentioned in a hard copy from the Kiosk system and the Kiosk system should be showing home page after delivering the hard copy.				

7.6 **Check Order List**

The test case is for the employee to see the orders comes from the Kiosk system.

Project Name: Smart Cafeteria of AIUB			Test Designed by: Mohammed Mofizur Rahman	
Test Case ID: SCA_06			Test Designed date: 15 June, 2022	
Test Priority (Low, Medium, High): Medium			Test Executed by: Md. Mridul Hosen	
Module Name: Check order list			Test Execution date: 17 June, 2022	
Test Title: Order List from the employee view with waiting time showing.				
Description: Test all the orders are visible or not				
Precondition (If any): User/Students must order something using the Kiosk system.				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Go to Employee page 2. Check order list.	Order something with the payment.	Employee should see all the products in details with the price which are ordered by the users	As expected,	Pass
Post Condition:				

7.7 **Show waiting time with confirmation message**

This Test case will test whether the employee can set waiting time for the users or not.

<table>
<tr><td colspan="3">Project Name: Smart Cafeteria of AIUB</td><td colspan="2">Test Designed by: Mohammed Mofizur Rahman</td></tr>
<tr><td colspan="3">Test Case ID: SCA_07</td><td colspan="2">Test Designed date: 15 June, 2022</td></tr>
<tr><td colspan="3">Test Priority (Low, Medium, High): Low</td><td colspan="2">Test Executed by: Md. Mridul Hosen</td></tr>
<tr><td colspan="3">Module Name: Show waiting time with confirmation message</td><td colspan="2">Test Execution date: 19 June, 2022</td></tr>
<tr><td colspan="3">Test Title: Showing waiting time to the user and confirmation message.</td><td colspan="2"></td></tr>
<tr><td colspan="3">Description: Test whether an employee can set waiting time for the user and give a confirmation message.</td><td colspan="2"></td></tr>
<tr><td colspan="3">Precondition (If any):</td><td colspan="2"></td></tr>
<tr><td>Test Steps</td><td>Test Data</td><td>Expected Results</td><td>Actual Results</td><td>Status (Pass/Fail)</td></tr>
<tr><td>1. Go to Employee page
2. Check order list.
3. Set a time
4. Send confirmation message</td><td></td><td>User should see a waiting time and a confirmation message.</td><td>As expected,</td><td>Pass</td></tr>
<tr><td colspan="3">Post Condition:</td><td colspan="2"></td></tr>
</table>

7.8 **Update Food Items and prices**

This Test case is for manager to add, delete and update items and their prices.

Project Name: Smart Cafeteria of AIUB	Test Designed by: Mohammed Mofizur Rahman
Test Case ID: SCA_08	Test Designed date: 16 June, 2022
Test Priority (Low, Medium, High): Medium	Test Executed by: Md. Mridul Hosen
Module Name: Update food items and prices	Test Execution date: 20 June, 2022
Test Title: Adding and updating the items and prices.	
Description: Test by adding and updating item list and the prices.	

Precondition (If any): manager should have registered to the system to change item list,				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Login to the system as a manager. 2. Click add option to add food item.	Add Takos and set price to 80 BDT	Takos should be added with the price showing to the user and stored in database.	As expected,	Pass
Post Condition: Database should be updated with Takos and its price.				

7.9 **Update Employee Database**

This Test case is for testing the employee details database for the manager.

Project Name: Smart Cafeteria of AIUB			Test Designed by: Mohammed Mofizur Rahman	
Test Case ID: SCA_09			Test Designed date: 17 June, 2022	
Test Priority (Low, Medium, High): Low			Test Executed by: Md. Mridul Hosen	
Module Name: Update employee database			Test Execution date: 20 June, 2022	
Test Title: Showing all the details of employees.				
Description: Test to change the details of an employee				
Precondition (If any):				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Login as a manager. 2. Go to employee list.	Add an employee with name: Aminul Islam, Gmail: aminul149@gmail.com address: Dhaka, blood group: A+.	The employee list should be updated showing Aminul Islam as a new employee.	As expected,	Pass
Post Condition: The employee list is updated.				

8. ITEM PASS/FAIL CRITERIA

Total 9 test cases were implemented for the Kiosk system. Firstly, when applied the test case to system at the unit testing, 85% of the test cases were passed successfully. But unfortunately, 15% were failed. The test cases were failing due to some query related issues on the database. When the test cases were applied after solving query related problem, all the test cases were successfully executed and thus passed the test.

9. TEST DELIVERABLES

- Test plan
- Test results documents
- Test summary
- Defect report
- Defect reports and summaries
- Screen prototypes

10. STAFFING AND TRAINING NEEDS

This section shows how to staff the test jobs and prepare them for the work. Staffing is set for the duration of the project. It's realistic to assume that the vast majority of the staff will agree to do some testing. The following occupations are recognized:

Project Manager: Responsible for maturing the complete execution of the Web website. This includes creating requirements, managing the seller relationship, and overseeing the testing cycle.

Test Manager: Responsible for fostering the expert test strategy, examining the test deliverable, dealing with test cycles, collecting measurements and reporting progress to the Project Manager, and recommending when testing should be completed.

Test Engineer: Planning tests, creating test methods, creating test information, running tests, preparing occurrence reports, examining episodes, writing mechanized test strategies, and detailing measurements to the test administrator are all responsibilities of this position.

The test manager and test specialists should be familiar with the website development life cycle. Because this project is being developed in a traditional manner, this is a nonexclusive depiction of Staffing and Training requirements. As a result, the names of conscious people for each project aren't given.

11. RESPONSIBILITIES

Name	Role	Responsibilities
Md. Abir Hossain	Project Manager	1. Requirement analysis and make plans for designing of the project. 2. The whole Designing of the Project. 3. Execute all the test cases and report defects. 4. Control whole project as designed.
Mohammed Mofizur Rahman	Quality Analyst	1. Creation of test plans, test forms, test cases and test information. 2. Carry out testing as per the characterized methods. 3. Prepare all reports related to program testing carried out.
Md. Mridul Hosen	Developer	1. Researching, designing, implementing, and managing software programs. 2. Writing and implementing efficient code. 3. Deploying software tools, processes, and metrics.
Mohammed Mofizur Rahman	Test Engineer	1. Check the characteristics of all the testing activities 2. Check all obligations of test planning 3. Prepare the report of testing activities. 4. Control of the test flow of Integration and System Testing. 5. Execute all the test cases. 6. Find out severe and Major bugs on the programs. 7. Ensure that the system is almost defect less (about 98%). 8. Make documentation.

12. TESTING SCHEDULE

Time has been allocated within the project plan for the following testing activities. The specific dates and times for each activity are defined in the project plan timeline. The persons required for each process are detailed in the project timeline and plan as well.

Task Name	Duration	Responsible
Requirement Analysis	4 days	Project Team
Design	8 days	Project Manager
Writing codes	22 days	Developers
Unit Test	20 days	Developers
Integration Test	12 days	Developer Team
System test	10 days	Project Manager & Test Engineer
Acceptance Test	8 days	Test Engineer & Potential Customers/ Users
Feedback	7 days	Potential Customers/ Users
Documentation	7 days	Test Engineer

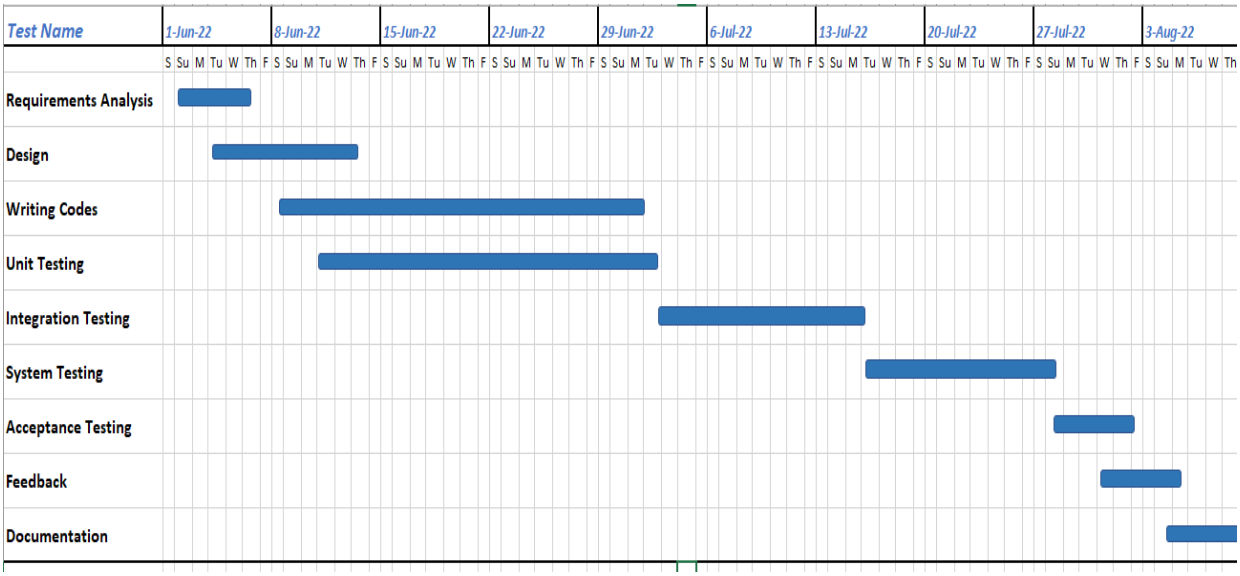

Fig.01: Test Schedule

13. PLANNING RISKS AND CONTINGENCIES

Since both the hardware and the software have to work together, the building of this Kiosk system may costly and the maintenance can be a hassle. Person to person interaction is very limited. Since it is self-service system, the user must be careful during the selection of food items.

14. APROVALS

Project Sponsor	American International University-Bangladesh(AIUB)
Development Management	Md. Mridul Hosen
EDI Project Manager	Mohammed Mofizur Rahman
RS Test Manager	Md. Mridul Hosen
RS Development Team Manager	Mohammed Mofizur Rahman
Reassigned Sales	Md. Abir Hossain
Order Entry EDI Team Manager	Md. Abir Hossain