

Dokan – An E-Commerce Site

Software Project-2

Submitted By

|  |  |
| --- | --- |
| 18-38441-2 | Asif Ahmed |
| 18-38454-2 | Shafiq Ahmed |
| 18-38341-2 | Tajbiur Shahrior Rimon |

Department of Computer Science

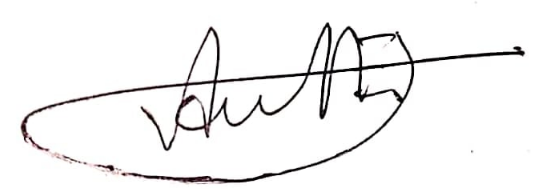
Faculty of Science & IT

American International University Bangladesh

June, 2022

|  |
| --- |
| Declaration |

We declare that this project is our original work and has not been submitted in any form for another degree or diploma at any university or other institute of tertiary education. Information derived from the published and unpublished work of others has been acknowledged in the text and a list of references is given.

|  |  |
| --- | --- |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  **Asif Ahmed**  18-38441-2  Software Engineering | \_\_\_\_\_\_\_\_\_\_\_\_\_\_  **Shafiq Ahmed**  18-38454-2  Software Engineering |
|  |  |
|  |  |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  **Tajbiur Shahrior Rimon**  18-38341-2  Computer Information System |  |

|  |
| --- |
| Approval |

The project titled “Dokan – An E-Commerce Site” has been submitted to the following respected members of the board of examiners of the department of computer science in partial fulfilment of the requirements for the degree of Bachelor of Science in Computer Science on /06/2022 and has been accepted as satisfactory.

|  |  |
| --- | --- |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  **MD. HASIBUL HASAN**  Assistant Professor & Supervisor  Department of Computer Science  American International University-Bangladesh | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  **ABHIJIT BHOWMIK**  Associate Professor & External  Department of Computer Science  American International University-Bangladesh |
|  |  |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  **DR. MD. ABDULLAH - AL – JUBAIR**  Assistant Professor & Head-In-Charge  Department of Computer Science  American International University-Bangladesh | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  **PROF. DR. TAFAZZAL HOSSAIN**  Pro-VC & Dean  Faculty of Science & Technology  American International University-Bangladesh |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  **DR. CARMEN Z. LAMAGNA**  Vice Chancellor  American International University-Bangladesh | |
| **Table of Contents** | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Chapter 1: Project Initiation** | | | | **7** | |
| 1.1 | | Objectives | |
| 1.2  1.3  1.4 | | Scopes  Abstract  Definitions, Acronyms, and Abbreviations | |
| **Chapter 2: Background Study**   |  |  | | --- | --- | | 2.1      2.2 | SDLC  2.1.1 Agile  2.1.2 Extreme Programming  Requirement Collection | | | | | **9** | |
| **Chapter 3: Requirement Specification** | | | | **13** | |
| 3.1 | | System Features. | |  | |
| 3.2 | | Functional Requirement | |  | |
| 3.3  3.4      3.5        3.6  3.7 | | Non-Functional Requirements  Functionality  3.4.1 Security  3.4.2 Usability  3.5.1 Understandability  3.5.2 Maintainability  3.5.2 Analyzability  Testability  Device Requirements  3.7.1 Hardware Requirements  3.7.2 OS Requirements | |  | |
| **Chapter 4: Data Requirements Specification.** | | | | **23** | |
| 4.1  4.2 | | Logical Data Model-UML diagrams  User Stories | |  | |

**Chapter 5: Impact Analysis 33**

**Chapter 6: Software Testing 34**

|  |  |
| --- | --- |
| 6.1  6.2  6.3 | Test Cases - Cashier  Test Cases – Admin  Test Cases - Accountant |

**Chapter 7: User Interface 46**

**Chapter 8: Conclusion and Future Works 73**

**References 74**

|  |
| --- |
| **List of Figures** |

|  |  |
| --- | --- |
| **Fig. 1** Entity-Relationship Diagram | 23 |
| **Fig. 2.1** Admin – Use Case Diagram | 24 |
| **Fig. 2.2** Cashier – Use Case Diagram | 25 |
| **Fig. 2.3** Accountant – Use Case Diagram  **Fig. 3.1** Admin – Activity Diagram  **Fig. 3.2** Cashier – Activity Diagram  **Fig. 3.3** Accountant – Activity Diagram | 26  27  28  29  30 |

|  |
| --- |
| **Chapter 1: Project Initiation** |

**1.1 Objective**

Online shopping is becoming popular with the time in Bangladesh. Many people prefer online shopping rather than going shopping malls. This project aims to give an e-commerce platform for both customers and vendors using React typescript and Java spring boot frameworks.

**1.2 Abstract**

Dokan – An E-Commerce Site is a multi-faceted, multi-centered web application which focuses on giving an e-commerce platform for both customers and vendors for trading. This will save their time and make their life easier.

There are three user classes in this project:

1. Admin (Maintainer and administer of the system)
2. Vendor (Sell the products)
3. Customer (Buy the products)

These three user classes will work in a simultaneous manner to keep the web application active and work properly.

**1.3 Scopes**

* Making online shopping more efficient
* Decreasing Fraud
* Decreasing workload
* Saving time
* Creating more efficient
* Smoothening the online shopping
* Introducing modern payment systems
* Introducing purchase protection
* Making life easier
* Making the system more reliable

**1.2 Definitions, Acronyms, and Abbreviations**

|  |  |
| --- | --- |
| **Tern** | **Definition** |
| **FR** | **Functional Requirement** |
| **QR** | **Quality Requirement** |
| **DESC** | **Description** |
| **DEP** | **Dependency** |
| **DKN** | **Dokan – An E-Commerce Site** |

|  |
| --- |
| **Chapter 2: Background Study** |

**2.1 SDLC**

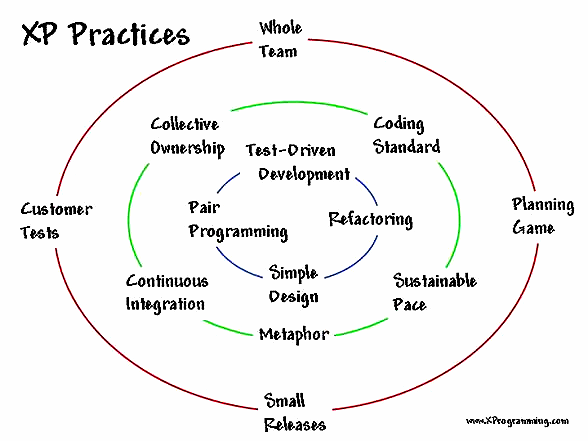
A systems development life cycle is composed of a number of clearly defined and distinct work phases which are used by systems engineers and systems developers to plan for, design, build, test, and deliver information systems. Like anything that is manufactured on an assembly line, an SDLC aims to produce high-quality systems that meet or exceed customer expectations, based on customer requirements, by delivering systems which move through each clearly defined phase, within scheduled time frames and cost estimates. Computer systems are complex and often (especially with the recent rise of service-oriented architecture) link multiple traditional systems potentially supplied by different software vendors. To manage this level of complexity, a number of SDLC models or methodologies have been created, such as waterfall, spiral, **Agile software development**, rapid prototyping, incremental, and synchronize and stabilize. (“Systems development life cycle,” 2021)

**2.1.1 Agile**

Agile software development refers to software development methodologies centered round the idea of iterative development, where requirements and solutions evolve through collaboration between self-organizing cross-functional teams. The ultimate value in Agile development is that it enables teams to deliver value faster, with greater quality and predictability, and greater aptitude to respond to change. Scrum and Kanban are two of the most widely used Agile methodologies.[2][3] However ***Extreme Programming***, also known as***XP*** will be used in this case.

**2.1.2 Extreme Programming**

Extreme Programming (XP) is an agile software development framework that aims to produce higher quality software, and higher quality of life for the development team. XP is the most specific of the agile frameworks regarding appropriate engineering practices for software development.



Summary of Extreme Programming. [Source: Infoq.com]

**Why XP?**

The reason Extreme Programming was chosen for this project is because of the state of dynamically changing software requirements, the size of the development team consisting on only 3 members located different parts of the country and its allowance of automated unit and functional tests which makes the development much faster than usual.

**2.2 Requirement Collection**

Requirement collection is the process of determining, documenting, and managing stakeholder needs & requirements to meet project objectives.

A simplistic yet detailed process was used when collecting requirements. The goal was to collect as much detail as possible and reduce technical jargons. A member of the team acted as Business Analyst and secured the requirements as successfully as possible.

The requirement collection process was started by “Data Collection” the procedures used were:

1. Surveys
2. Interviews
3. Focus groups
4. Brain Storming
5. Benchmarking

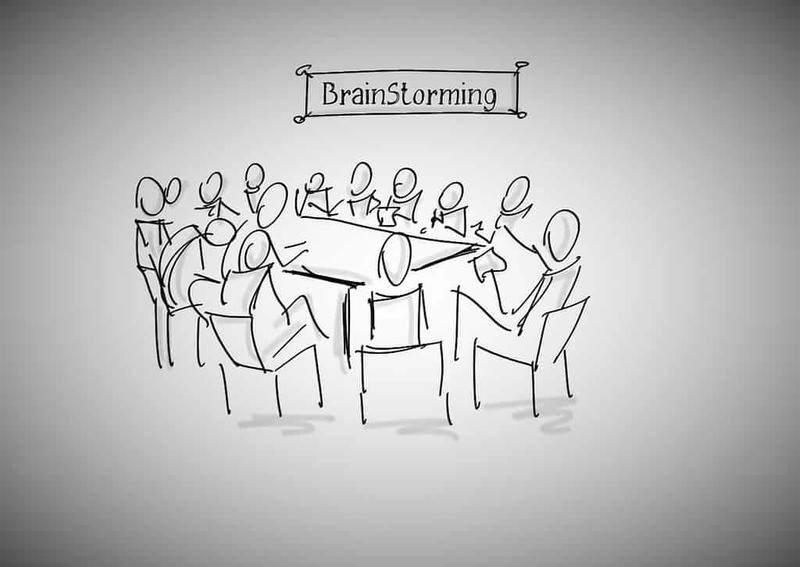


Image Source: Flickr

The second part of Requirement Collection is Data Analysis, also known as Document Analysis.

Data Analysis focuses on policies, procedures, use cases, market literature etc.

The next step is one of the most important ones. It’s where the Decisions are made. The processes used to make the decisions are given below.

1. Multi-Voting. This follows multiple criteria. Unanimous, Majority and Plurality.
2. Autocratic
3. Multi-criteria

Next part is the data representation. At first ER Diagram was used primarily. Affinity diagram wasn’t used because the number of stakeholders is low in this case and it would only make the requirements more verbose. Interactions however were documented using an activity diagram.

A functional prototype was also created to get realistic feedback based on the prototype. Positive feedbacks were given a green light the red ones were sent for further reviews.

**The Delphi Technique**

further The Delphi Technique is used to realistically absorb the needs and requirements from the stakeholders without any bias infecting it. Everything is done through anonymously and are analyzed thoroughly. After the results are out, they are shared among the stakeholders. This reduces the bias significantly compared to other processes.

|  |
| --- |
| **Chapter 3: Requirements Specification** |

**3.1 System Features**

**Admin’s Role:**

1. Admin will monitor system data (total users, sales etc.).
2. Admin will manage users.
3. Admin will add categories.
4. Admin will manage categories.
5. Admin will manage orders.
6. Admin can search products, users and categories.
7. Admin will monitor user reviews.
8. Admin can update own profile.
9. Admin must login in order to perform his activity.
10. After completing activities, admin can log-out from the system.

**Vendor’s roles:**

1. Vendor will register using proper information and through email verification.
2. Vendor will log in with username and password.
3. A dashboard with all products of vendor’s shop will appear.
4. Vendor will add/update products.
5. Vendor will manage products.
6. Vendor will monitor user reviews.
7. Vendor can update own profile.
8. Vendor can update shop profile.
9. Vendor can search products.
10. After completing activities, vendor can log-out from the system.

**Customer’s roles:**

1. Customer will register using proper information and through email verification.
2. Customer will log in with username and password.
3. A dashboard with all products will appear.
4. Customer can add/update products in the shopping cart.
5. Customer can place orders.
6. Customer can add/remove products in the Wishlist.
7. Customer can update own profile.
8. Customer can view product reviews.
9. Customer can search products.
10. Customer can add/edit reviews after shopping.
11. After completing activities, customer can log-out from the system.
    1. **Functional Requirements**

This section consists of the functional requirements. In software engineering and systems engineering, a functional requirement defines a function of a system or its component, where a function is described as a specification of behavior between inputs and outputs. [3]

There are three user classes in this software. They are Admin, Vendor and Customer.

**User Class 1 – Admin**

**TITLE: Login-Admin**

ID: FR1

DESC: Admin should be able to login with appropriate credentials.

DEP: Nil

**TITLE: Monitoring-Data**

ID: FR2

DESC: Admin should be able to monitor all system data

DEP: FR1

**TITLE: User-Management**

ID: FR3

DESC: Admin should be able to manage all the users

DEP: FR1

**TITLE: Add-Category**

ID: FR4

DESC: Admin should be able to add new categories

DEP: FR1

**TITLE: Manage-Category**

ID: FR5

DESC: Admin should be able to manage categories.

DEP: FR4

**TITLE: Searching**

ID: FR6

DESC: Admin should be able to search users, categories and products.

DEP: FR1

**TITLE: Order-Management**

ID: FR7

DESC: Admin should be able to manage orders

DEP: FR1

**TITLE: Monitoring-Reviews**

ID: FR8

DESC: Admin should be able to monitor user reviews.

DEP: FR1

**TITLE: Profile-Update**

ID: FR9

DESC: Admin should be able to update his/her own profile.

DEP: FR1

**TITLE: Logout-Admin**

ID: FR10

DESC: Admin should be able to logout.

DEP: FR1

**User Class 2 – Vendor**

**TITLE: Registration-Vendor**

ID: FR11

DESC: Vendor should be able to register with appropriate information via email verification.

DEP: Nil

**TITLE: Login-Vendor**

ID: FR12

DESC: Vendor should be able to login with appropriate credentials.

DEP: FR11

**TITLE: Add-Products**

ID: FR13

DESC: Vendor should be able to add new products.

DEP: FR12

**TITLE: Update-Products**

ID: FR14

DESC: Vendor should be able to update products.

DEP: FR12, FR13

**TITLE: Manage-Products**

ID: FR15

DESC: Vendor should be able to manage products.

DEP: FR12, FR13

**TITLE: Update-Profile**

ID: FR16

DESC: Vendor should be able to update own and shop profile

DEP: FR11, FR12

**TITLE: Monitoring-Reviews**

ID: FR17

DESC: Vendor should be able to monitor user reviews of shop’s products.

DEP: FR12, FR13

**TITLE: Search-Products**

ID: FR18

DESC: Vendor should be able to search shop’s products.

DEP: FR12, FR13

**TITLE: Logout-Vendor**

ID: FR19

DESC: Vendor should be able to logout.

DEP: FR12

**User Class 3 – Customer**

**TITLE: Registration-Customer**

ID: FR20

DESC: Customer should be able to register with appropriate information via email verification.

DEP: Nil

**TITLE: Login-Customer**

ID: FR21

DESC: Customer should be able to login with appropriate credentials.

DEP: FR20

**TITLE: Add/Update-Cart**

ID: FR22

DESC: Customer should be able to add/update products in shopping cart.

DEP: FR21

**TITLE: Place-Order**

ID: FR23

DESC: Customer should be able to place orders.

DEP: FR21, FR22

**TITLE: Add/Update-Wishlist**

ID: FR24

DESC: Customer should be able to add/update products in Wishlist.

DEP: FR21

**TITLE: Searching**

ID: FR25

DESC: Customer should be able to search products and view product details.

DEP: FR21

**TITLE: Review**

ID: FR26

DESC: Customer should be able to add/update product review.

DEP: FR21, FR23

**TITLE: Profile-Update**

ID: FR27

DESC: Customer should be able to update own profile.

DEP: FR21

**TITLE: Logout-Customer**

ID: FR28

DESC: Customer should be able to logout.

DEP: FR21

* 1. **Non-Functional Requirements**

In systems engineering and requirements engineering, a non-functional requirement (NFR) is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviors. They are contrasted with functional requirements that define specific behavior or functions. These can include Usability, Interoperability. Security, understandability etc.

* 1. **Functionality**

DKN will create a platform for both shop owners and customers, where vendors will sell their products and customer will buy products and admin will manage the whole system.

* + 1. **Security**

Multiple security protocols have been used for DKN. Here are some of them:

* + - 1. Passwords are hashed with CRC256.
      2. Database is stored on MySQL.
      3. The whole system is protected with SSL.
      4. Vendors are verified to avoid scams.
      5. SQL Injection protection has been added,
  1. **Usability**

DKN has been created with modern UX practices which came up with decades of Human Computer Interaction research. It is made to be intuitive, manual-free. Ease of use one of the first priorities of DKN.

* + 1. **Understandability**

Intuitive interface makes the understandability of DKN a much higher percentile than usual. The modern UX design implementation removes the jargon other software usually contains.

* + 1. **Maintainability**

Automatic error generation feature makes the maintenance process much smoother than usual.

* + 1. **Analyzability**

Trusted analytical tools such as google analytics has been embedded into TMS. There are also built-in tools to generate logs of every process in the system.

* 1. **Testability**

Testing is a process to examine whether the expected requirement is similar to the final result and to make sure that the production build is free from bugs, glitches etc.

As per ANSI/IEEE 1059, Testing in Software Engineering is a process of evaluating a software product to find whether the current software product meets the required conditions or not. The testing process involves evaluating the features of the software product for requirements in terms of any missing requirements, bugs or errors, security, reliability and performance.

There are three types of testing:

Functional testing

Performance Testing or Non-Functional testing

Maintenance Testing

As XP (Extreme Programming) has been used to develop DKN. Test-Driven Development has been followed. Here are the tests that have been applied to DKN:

**Unit Testing:**  Unit Testing is based on units. These units are small part of codes that acts as an individual part when testing is applied to it. It is done programmers.

**Integration Testing:**  It focuses on the construction and design of the software. You need to see that the integrated units are working without errors or not.

**System Testing:**  In this method, the software is compiled as a whole and then tested as a whole. This testing strategy checks the functionality, security, portability, amongst others.

* 1. **Device Requirements**
     1. **Hardware Requirements**

**Minimum Requirements:**

* Processor: Intel Celeron @ 1.4GHZ
* Memory: 4GB
* Graphics Card: NVIDIA 6800 GT 1GB / AMD HD 3870 1GB (DX 9, 10, 10.1, 11)
* Storage Space: 5GB
  + 1. **OS Requirements**

Windows 8.1 64 Bit

Windows 8 64 Bit

Windows 7 64 Bit Service Pack 1

Windows 10 64 bit (All versions)

|  |
| --- |
| **Chapter 4: Data Requirements Specification** |

* 1. **Logical Data Model-UML Diagrams**

Fig 1: Entity-Relationship Diagram

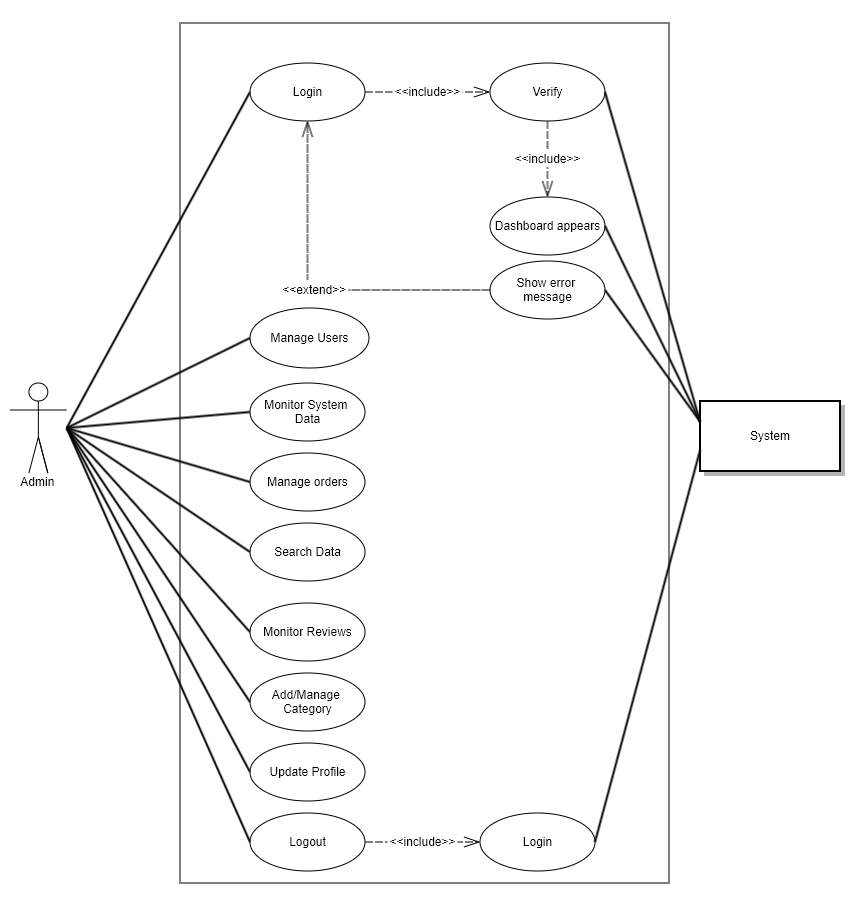


Fig 2.1: Use Case Diagram-Admin

Fig 2.2: Use Case Diagram-Vendor

Fig 2.3: Use Case Diagram-Customer

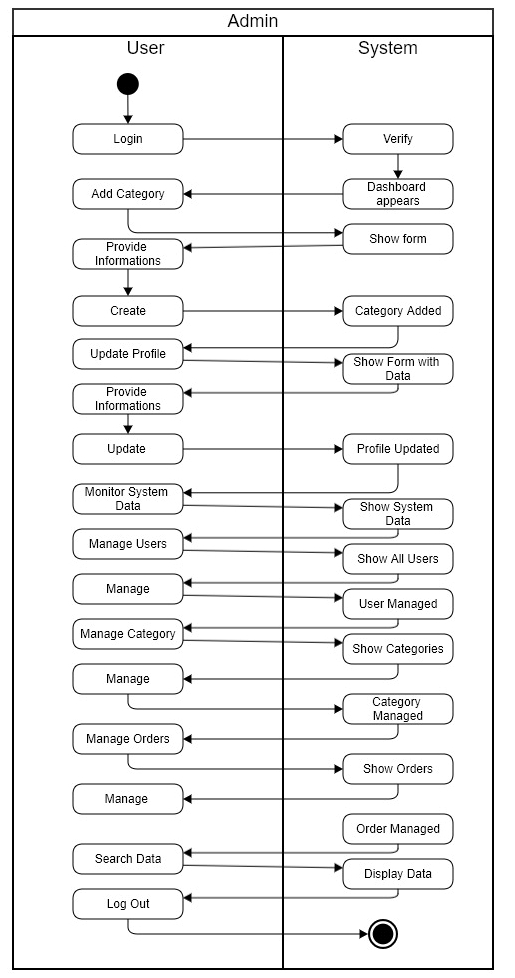


Fig 3.1: Activity Diagram-Admin

Fig 3.2: Activity Diagram-Vendor

Fig 3.3: Activity Diagram-Customer

* 1. **User Stories**
     1. Admin

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **FR-TITLE** | **As a/an** | **I want to** | **So that** |
| **1** | Login-Admin | Admin | Log-in to DKN | I can administer |
| **2** | Monitoring-Data | Admin | Monitor Data | I have the system overview. |
| **3** | User-Management | Admin | Manage Users | I can manage the users. |
| **4** | Add-Category | Admin | Add Categories | I can add new categories |
| **5** | Manage-Category | Admin | Manage Category | I can manage categories |
| **6** | Searching | Admin | Search | I can find data. |
| **7** | Order-Management | Admin | Manage Orders | I can manage orders |
| **8** | Monitoring-Reviews | Admin | Monitor Reviews | I can monitor all the reviews |
| **9** | Profile-Update | Admin | Update Profile | I can update my own profile |
| **10** | Logout-Admin | Admin | Logout | My portal remains secured |

* + 1. Vendor

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **FR-TITLE** | **As a/an** | **I want to** | **So that** |
| **1** | Registration-Vendor | Vendor | Register into DKN | I can login as a vendor. |
| **2** | Login-Vendor | Vendor | Log-in to DKN | I can access dashboard |
| **3** | Add-Products | Vendor | Add Products | I can add new products |
| **4** | Update-Products | Vendor | Update Products | I can update existing products. |
| **5** | Manage-Products | Vendor | Manage Products | I can manage existing products |
| **6** | Update-Profile | Vendor | Update Profile | I can update my profile |
| **7** | Monitoring-Reviews | Vendor | View Reviews | I can check reviews of my products |
| **8** | Search-Products | Vendor | Search | I can find my products |
| **9** | Logout-Vendor | Vendor | Logout | My portal remains secured |

* + 1. Customer

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **FR-TITLE** | **As a/an** | **I want to** | **So that** |
| **1** | Registration- Customer | Customer | Register into DKN | I can login as a customer. |
| **2** | Login- Customer | Customer | Log-in to DKN | I can access dashboard |
| **3** | Add/Update-Cart | Customer | Add/Update Cart | I can add new products  in cart or update cart |
| **4** | Place-Order | Customer | Place Order | I can place orders |
| **5** | Add/Update-Wishlist | Customer | Add/Update-Wishlist | I can add new products  in Wishlist or update Wishlist |
| **6** | Update-Profile | Customer | Update Profile | I can update my profile |
| **7** | Searching | Customer | Search | I can find products |
| **8** | Review | Customer | View Reviews | I can view product reviews |
| **9** | Logout- Customer | Customer | Logout | My portal remains secured |

|  |
| --- |
| **Chapter 5: Impact Analysis** |

This project was designed to exhibit immense amount of impact and we believe it’s impact will be noteworthy. Here’s our prediction how DKN will have an impact:

* 1. DKN shall make online shopping easier.
  2. DKN shall save the time of people.
  3. DKN shall decrease fraud on online shopping by verifying vendors.
  4. DKN shall make the business easier for vendors.
  5. DKN shall save money of the vendors as they don’t need big showrooms or more staffs for business.
  6. DKN shall make online shopping more efficient.

In any case, the impact DKN will have on the online shopping will be immense and it may change the way how we view online shopping in Bangladesh.

|  |
| --- |
| **Chapter 6: Software Testing** |

|  |
| --- |
| **Chapter 7: User Interface** |

|  |
| --- |
| **Chapter 8: Conclusion and Future Works** |

Online shopping has become a daily necessity for people. The economy of a country largely depends on it nowadays. The objective of this project is to give an online platform for easier and efficient shopping for customers and business platform for vendors.

The automation, efficiency, interaction DKN provides is not just satisfactory but we predict it could revolutionize the industry the country abhors so much.

The future works planned by the same team:

1. Add more features in DKN
2. Make DKN more efficient
3. TMS-Android
4. TMS-IOS

|  |
| --- |
| **References** |

[1] “Extreme Programming (XP)” <https://www.agilealliance.org/glossary/xp/>

[2] “Agile 101” <https://www.agilealliance.org/agile101/>

[3] “The Top 7 Popular Agile Development Approaches” <https://www.visual-paradigm.com/scrum/the-top-7-popular-agile-development-approaches/>

[4]” Systems Development Life Cycle” <https://en.wikipedia.org/wiki/Systems_development_life_cycle>