**ONLINE SHOPPING SYSTEM**

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **S.NO** | **HEADINGS** | **PAGE NO.** |
| 1 | Goals and Exit Criteria | 3-4 |
| 2. | Items to Be Tested/Inspected | 5-6 |
| 3. | Test Process/ Methodologies | 6-10 |
| 4. | Resources | 11-12 |
| 5. | Schedule | 12-13 |
| 6. | Risks | 13 |
| 7. | Major Test Scenarios and Test Cases | 13-14 |

# **A. GOALS AND EXIT CRITERIA**

By achieving the following goals of we ensure very accurate performance with the highest quality product of our ecommerce application.

1. **Quality Goals:**

**i. Correctness:**

We display the products based on the category selected by the user. Hence the probability of user getting the product that he has searched for is high if our product database is huge.

The quality of the search feature can be improved gradually as more and more products are added to our database.

**ii. Budget Friendly:**

During the initial stages we would need less server capacity and less storage space since the range of products is less during this time. Moving forward we might require load balancers to balance the load and more storage space. But these costs can be covered as the number of products being sold increases hence pushing us to profit margin.

**iii. Usability:**

The User interface of our application is like all major ecommerce websites. Also, online shopping has gained tremendously since the onset of the pandemic we can say that most of the users have basic idea on how to search for a product and place the order.

**iv. Efficiency**: Our application consumes only little RAM space so we can say that it is efficient.

**v**. **Security**: Since we store sensitive information such as user’s address, phone number we need to consider the security of the application with utmost care. For achieving this we are encrypting and decrypting for all the http requests. Also moving forward, we are considering getting the https certificate for our domain name.

**vi. Functionality:** As mentioned earlier our application is like all major ecommerce websites so navigating between different pages is simple and easy.

1. **Robustness:**

Since this is an ecommerce application the load that we might receive during peak times such as during seasonal sales could be very high. We need to make sure that our system would run seamlessly during that time and ensure that no deadlocks are formed on the DB.

We can divide the incoming load to the server by using load balancers. We have built a scalable database model so that we would not face any issues once more and more products are added to the database.

**c. Schedule Goals:**

|  |  |
| --- | --- |
| **Project Phase** | **Goal** |
| Planning | Laying out a detailed plan of the project goals and breaking down the tasks |
| Design | Design the Database model and UI for the web application |
| Development (Phase 1) | Work on front end and backend services and integrate both. |
| Testing (Phase 1) | User acceptance testing of the product |
| Development (Phase 2) | Fix the bugs found in the previous phase |
| Testing (Phase 2) | Test the new release for any news bugs or discrepancies |
| Deployment | Deploy the application on the server and monitor |

**d. Performance Goals:**

The performance of the application depends on how quickly the search results are displayed to the user. To achieve this, we are avoiding using Joins while writing the queries in the backend services. Also, we are maintaining indexes on the tables with product information to reduce the query response time.

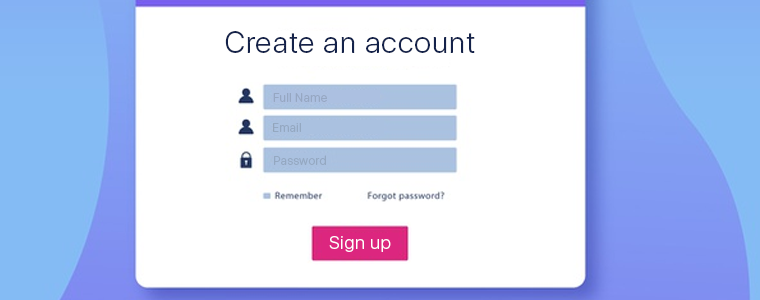
# **B. Items to be Tested/Inspected**

**UI Modules**: We have the following module in the UI testing:

**Sign up page**:

This page takes the basic details from the user such as email, phone no and password of his choice.

The same details are stored in the database with the password encrypted and the user is assigned a special ID which would be used in all the services that are related to the customer.



**Login Page**: This page will check if the user is already present in our database. If the user is present in our database, we will take him/her to the home page. If there is no record for the email entered by the user, we redirect to the registration page.

Graphical user interface

Description automatically generated

**Home Page**: Home page consists of all the ongoing sales details, previous order details. We need to make sure all the information displayed in the home matches the data that in our database and it is up to date as per the user’s activity.

**Products Page**: We display the details of the product that user has searched for. We need to assert that the correct product is displayed and all other details such as whether if it is available in the user’s pin code and all are also asserted.

Graphical user interface, website

Description automatically generated

**My Orders Page**: This page displays all the order place by the user along with product details and product quantity and status of the order. We need to assert that all the details displayed are correct.

**C. TEST PROCESS/ METHODOLOGIES**

**Black-box testing**:

|  |  |
| --- | --- |
| Test Case Number | 01 |
| Test Case Name | Boundary condition |
| Expected Results | Black box testing is a software testing method which is a critical element of assurance software quality that represents the main study of specifications, design, and coding. Increased visibility (capabilities) of the software as a system element and the "costs" that arise due to device failure lenient, motivates good planning through its testing thorough. Black box testing techniques focus on the information domain of the software, by doing a test case by partitioning the input domain of a program in a way that provides in-depth testing coverage. The Black Box itself has several methods in testing. Namely, the test method Graph-based explores the relationship between objects and behavior program. The equivalent partition divides the input domain into data classes it is possible to perform certain software functions. Boundary value analysis check the program's ability to handle data to the extent it can received. |

**Unit Testing:**

|  |  |
| --- | --- |
| Test Case Number | 02 |
| Test Case Name | Testing and debugging |
| Expected Results | In white box testing knowing the internal working of the product, tests can be conducted to ensure that internal operations are performed according to specification and all internal components have been adequately exercised. In white box testing logical path through the software are tested by providing test cases that exercise specific sets of conditions and loops. Using white box testing software developer can derive test case that • Guarantee that all independent paths within a module have been exercised at least once. • Exercise all logical decisions on their true and false side. • Exercise all loops at their boundaries and within their operational bound. • Exercise internal data structure to ensure their validity. At every stage of project development, I have tested the logics of the program by supplying the invalid inputs and generating the respective error messages. All the loops and conditional statements are tested to the boundary conditions and validated properly. |

**White-box testing:**

|  |  |
| --- | --- |
| Test Case Number | 03 |
| Test Case Name | Control Path Testing |
| Expected Results | In white box testing knowing the internal working of the product, tests can be conducted to ensure that internal operations are performed according to specification and all internal components have been adequately exercised. In white box testing logical path through the software are tested by providing test cases that exercise specific sets of conditions and loops. Using white box testing software developer can derive test case that • Guarantee that all independent paths within a module have been exercised at least once. • Exercise all logical decisions on their true and false side. • Exercise all loops at their boundaries and within their operational bound. • Exercise internal data structure to ensure their validity. At every stage of project development, I have tested the logics of the program by supplying the invalid inputs and generating the respective error messages. All the loops and conditional statements are tested to the boundary conditions and validated properly. |

**Acceptance Testing:**

|  |  |
| --- | --- |
| Test Case Number | 04 |
| Test Case Name | Acceptance Testing |
| Expected Results | User Acceptance Testing (UAT) involves testing an app, website, IoT, or other form of software to assess if it operates according to its functional requirements.  Although there are many kinds of tests that fit into the UAT framework, the main goal remains is to verify that the app, website, connected device, etc., meets the requirements necessary for release. With acceptance testing, brands can uncover anomalies and perform bug fixes that would have ultimately hindered the site and soiled your brand’s image. |

**D. RESOURCES**

1. **People:**

This team in total consists of five people, each person is associated with the below mentioned skills.

|  |  |
| --- | --- |
| Team Member | Skills |
| Nivas Reddy | HTML  CSS |
| Keerthana Sankaramaddi | Java Script  MySQL |
| Gayatri Mudimbi | HTML  CSS |
| Asha Jyothi Mannem | Java Script  MySQL |

**b. Testing System:**

Test execution platform for this project will be GitHub because GitHub has records of our previous versions i.e., it has version control. GitHub is fast and multiple people can work on that at the same time.

The testers of the project will report all the bugs they incur on a team document in Google Docs.

# **E. SCHEDULE:**

**i. Test-case development:**

During phase of 1:testing, test cases are created. Because the test cases are comprehensive, they are carried forward to phase 2: testing. New test cases may be added if necessary during phase 2: testing.

**ii. Test execution:**

According to the SDLC for this project, testing is completed in two stages. The test cases developed in phase 1: testing will be executed in both phases 1: testing and 2: testing.

**iii. Problem reporting and fixing:**

The errors discovered during the first two phases of testing will be addressed in the third phase. As a result, the errors discovered during phase 1: testing will be addressed during phase 2: development. Errors discovered during phase 2: development will be addressed during the final changes.

**F. RISKS:**

* Some websites don't encrypt data. They operate on outdated SSL certificates and HTTP protocols which make them vulnerable to attacks. Any website whose URL address starts with HTTP instead of HTTPS isn't secured, that's a red flag.
* One of the most common security risks of online shopping is identity theft. Cybercriminals can hack into private websites and steal users’ login credentials or credit card details.
* Not all e-commerce sites are genuine. Some are managed by cybercriminals with the intent to steal your money and personal information.
* Using unsecured WiFi for online shopping isn't such a good idea as it allows the movement of data across networks without encryption. It creates an opportunity for hackers to position themselves between you and the connection point.

**Solutions:**

* Keep your devices and operating system clean by updating the current version of apps and software running on them. These updates protect internet-connected devices from vulnerabilities and online attacks.
* Built with an advanced algorithm, the software detects malicious activities. It even goes as far as preventing you from unknowingly logging into a malicious website.
* Using a credit card to shop online is much safer than using a debit card. And that's because a credit card has more consumer protection if something goes wrong.

# **G. MAJOR TEST SCENARIOS AND TEST CASES:**

**Black box Testing:**

Black box testing is a software testing process in which the tester has no experience of the object being tester’s internal configuration, architecture, or execution. Black box testing is also called functional testing that ignores all internal mechanism in the system. I have decided to perform equivalence partitioning and boundary value analysis for the Online Shopping system. Black box testing involves testing a system with no prior knowledge of its internal workings. A tester provides an input, and observes the output generated by the system under test. This makes it possible to identify how the system responds to expected and unexpected user actions, its response time, usability issues and reliability issues.

**White Box Testing:**

White Box Testing In white box testing knowing the internal working of the product, tests can be conducted to ensure that internal operations are performed according to specification and all internal components have been adequately exercised. In white box testing logical path through the software are tested by providing test cases that exercise specific sets of conditions and loops. Using white box testing software developer can derive test case that

• Guarantee that all independent paths within a module have been exercised at least once.

• Exercise all logical decisions on their true and false side.

• Exercise all loops at their boundaries and within their operational bound.

• Exercise internal data structure to ensure their validity.

At every stage of project development, I have tested the logics of the program by supplying the invalid inputs and generating the respective error messages. All the loops and conditional statements are tested to the boundary conditions and validated properly.

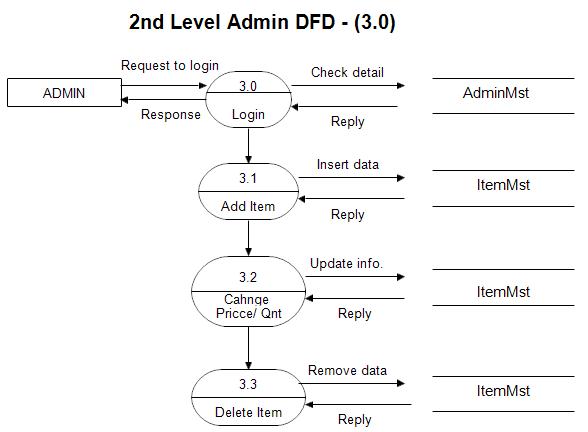
**Data Flow:**

1st Level Admin  Side DFD

The admin side DFD describe the functionality of Admin, Admin is an owner of the website. Admin can first add category of item and then add items by category wise. and admin can manage order and payment detail.

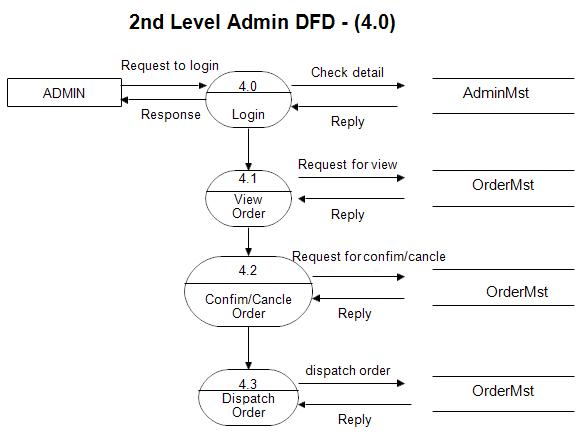
[](https://meeraacademy.com/wp-content/uploads/2016/09/adminside-first.jpg)

### 2nd Level – Admin side DFD (3.0)

[](https://meeraacademy.com/wp-content/uploads/2016/09/admin3.0.jpg)

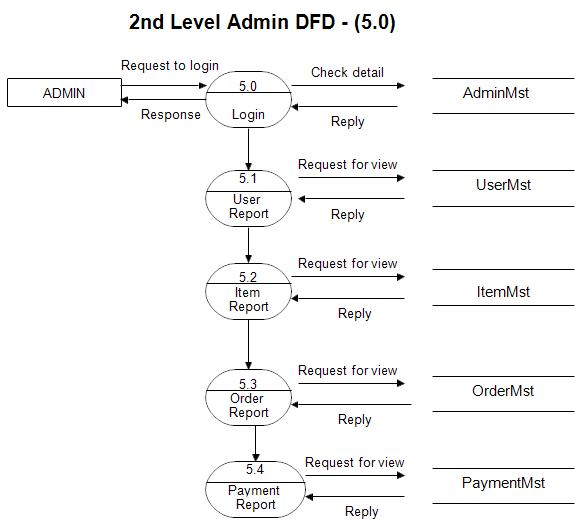
DFD for online shopping website project

### 2nd Level – Admin side DFD (4.0)

[](https://meeraacademy.com/wp-content/uploads/2016/09/admin4.0.jpg)

Admin side DFD for online shopping website project

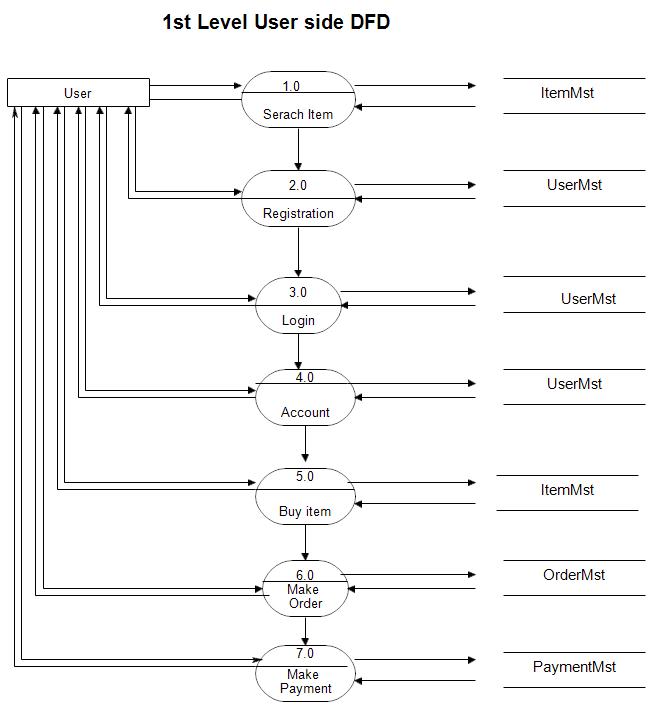
### 2nd Level – Admin side DFD (5.0)

[](https://meeraacademy.com/wp-content/uploads/2016/09/admin5.0.jpg)

Admin side DFD for online shopping website project

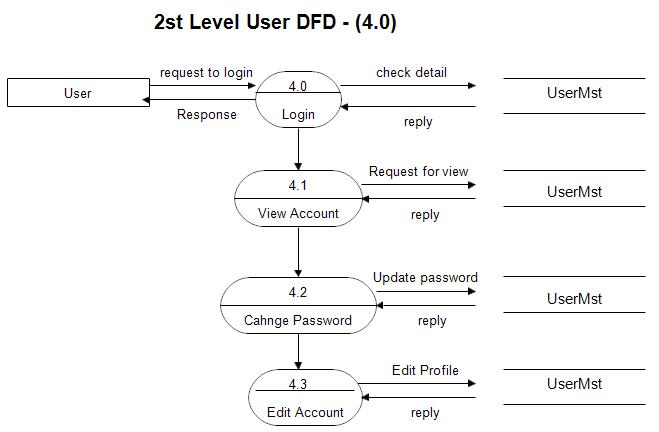
**1st level – User side Data flow Diagram**

The user is all people who operate or visit our website. User is a customer of a website. User can first select product for buy user must have to register in our system for purchase any item from our website. after register he can login to site and buy item by making online payment through any bank debit card or credit card.

[](https://meeraacademy.com/wp-content/uploads/2016/09/user1st.jpg)

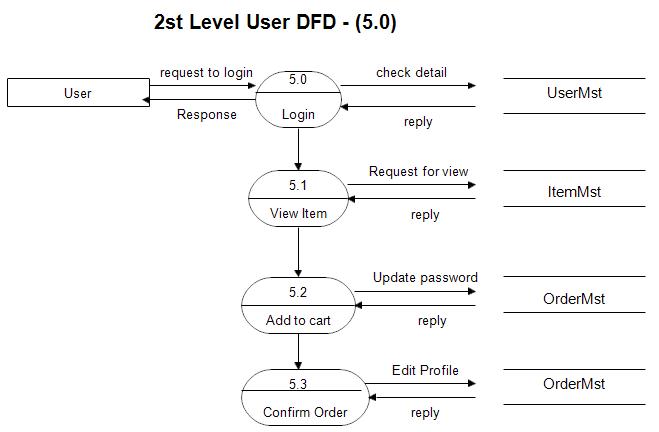
User side DFD for online shopping website project

### 2nd level – User side DFD (4.0)

[](https://meeraacademy.com/wp-content/uploads/2016/09/usr4.0.jpg)

User side DFD for online shopping website project

**2nd level – User side DFD (5.0)**

[](https://meeraacademy.com/wp-content/uploads/2016/09/usr5.0.jpg)

DFD for online shopping website project

The user is all people who operate or visit our website. User is a customer of a website. User can first select product for buy user must have to register in our system for purchase any item from our website. after register he can login to site and buy item by making online payment through any bank debit card or credit card.