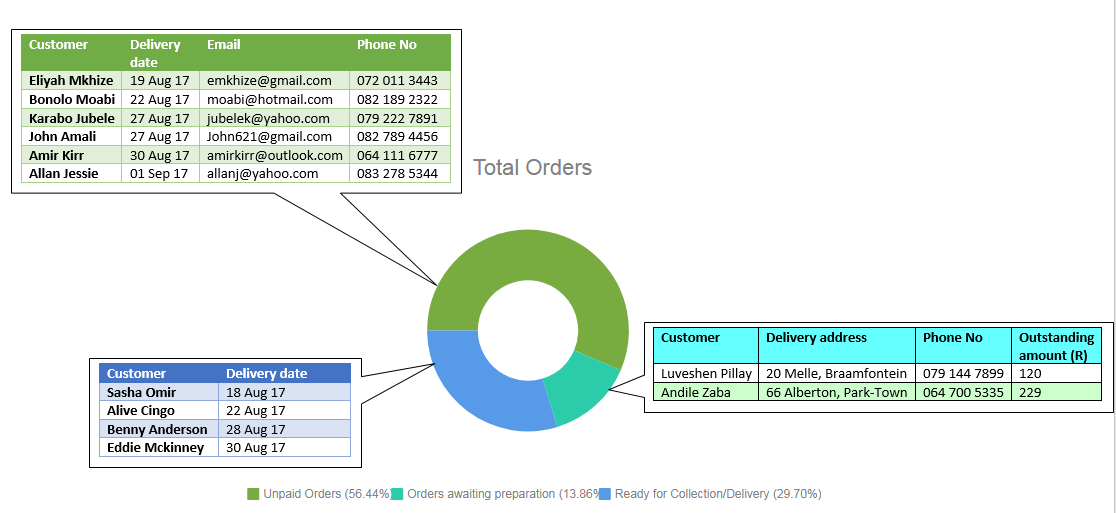
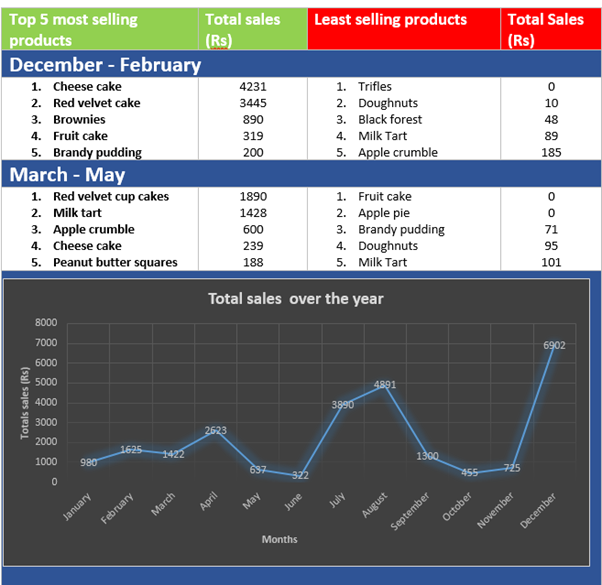
**Report 1:**

|  |  |
| --- | --- |
| **Report Name** | Meal Popularity Report |
| **Report Type** | Detailed yearly report |
| **Report Format** | Table listing the top 5 Meals which are most popular and top 5 Meals that are least popular. |
| **Report Recipient** | Manager |
| **Report Frequency** | Monthly |
| **Report justification/rationale** | Many Meals that are on offer are in higher or lower demand during different times of the year. For example, in November/December many corporate companies order savoury platters for the year-end functions. Also during religious months of fasting, her demand is lower as her customers don’t order many Meals.  Due to the nature of the client’s business, many Meals and the ingredients the manager buys rely on seasonality.   1. Due to this, a report that shows the most and least popular Meals for different time periods can help her determine which Meals are in demand and which Meals she needs to produce more during those time periods and make those Meals available to customers. 2. It also can help her determine she needs to stock up on to make those popular Meals 3. She may need to hire extra help if demand increases during busy periods. 4. This report will show consumer demand which will help her to modify or replace Meals that are losing popularity to appeal to current market trends. |
| **Decision(s) as a result of the report** | Decision on justification 1:  1.1 Increase available quantity for the Meals  Decision on justification 2:  2.1. Prepare better for an influx of orders during busy periods by buying more ingredients that make up those popular Meals  Decision on justification 3:  3.1 Replace Meals that customers are no longer interested in with something different.  Decision on justification 4:  4.1 Modify Meals that are growing less popular based on what is popular. |

**Report 1 Design:**



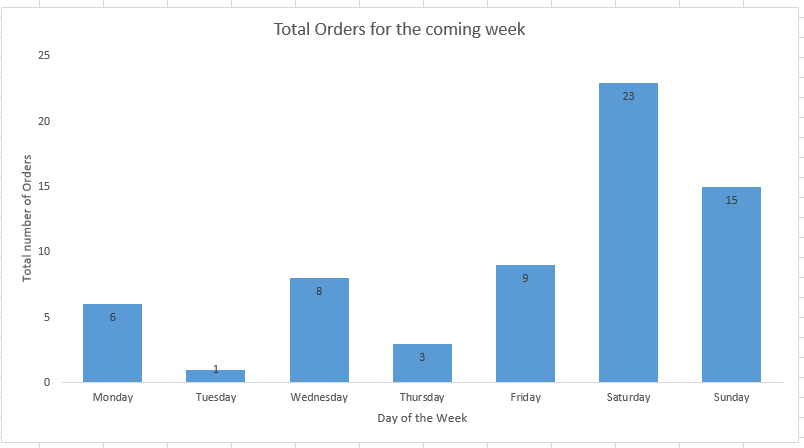
**Report 2:**

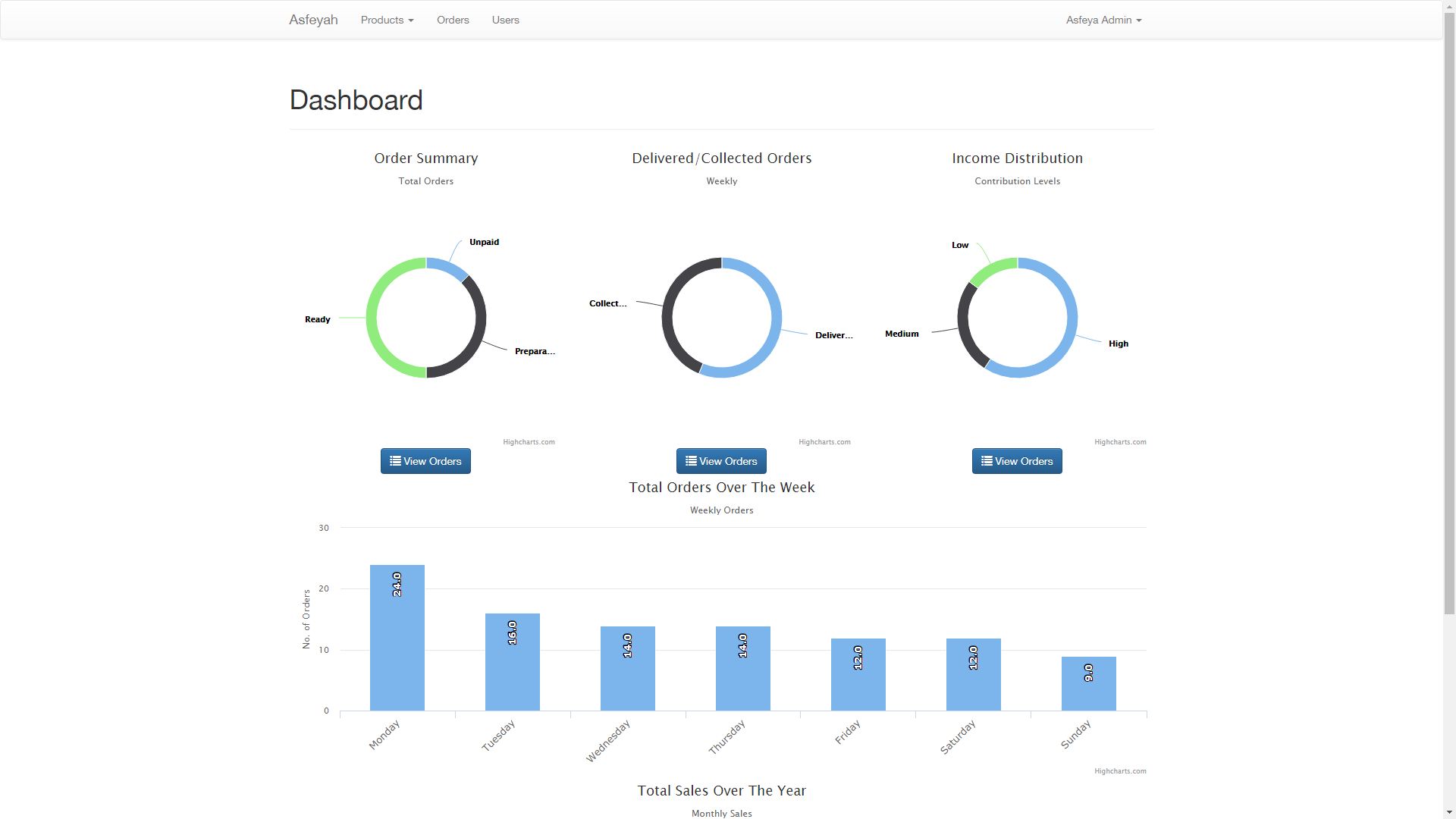
|  |  |
| --- | --- |
| **Report Name** | Sales over the year |
| **Report Type** | Summarised yearly report |
| **Report Format** | Bar graph with a trend line showing the total sales for each month |
| **Report Recipient** | Manager |
| **Report Frequency** | Yearly |
| **Report justification/rationale** | 1. This report tracks the sales made for the year. The report is mainly needed to monitor her performance over the year and also coincides with product popularity report to show demand during the year. 2. Our client also makes a lot of overseas trips during the year and this report can help her forecast off-peak and peak business times during the year so that she can know when to plan her trips and organise her schedule accordingly |
| **Decision(s) as a result of the report** | Decision(s) on justification 1:   * 1. Increase advertising and provide promotions for Meals   Decision on justification 2:   * 1. Plan her travel schedule according to off-peak and peak times |

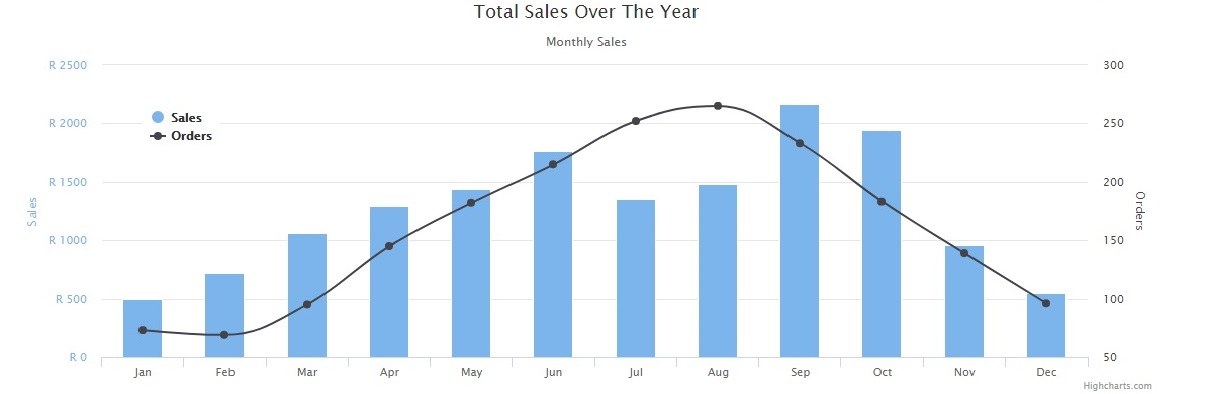
**Report 2 Design:**

**Report 3:**

|  |  |
| --- | --- |
| **Report name:** | Total Orders placed for the coming week |
| **Report Type:** | Summary |
| **Report Format:** | Bar graph showing total number of meals to be ordered for each day in the coming week |
| **Report Frequency:** | Real time |
| **Report justification/rationale** | This report will show total of placed orders that are to be delivered on each day for the coming week. From this the administrator will be able to see days in which she has the highest demand and plan her order preparation accordingly and may decide if she should hire additional help (this is something she currently does when she has too much demand, this report will inform that decision).  This report will also allow her see days on which she can accept special orders i.e. orders that are pre-ordered on Meals that not on the list of Meals available for that week.  This report could also be used in conjunction with another report we are planning to implement (Product popularity report) which shows the top 5 most ordered meals. She may look at the days that have the lowest number of orders and make the least selling Meals on special for that day.  She may also view the most ordered product and decide if she should increase available quantity and reduce available quantity of a product that is not ordered that much. |
| **Decisions:** | 1. Decide if she should hire additional help for some days in the week 2. Plan her preparation schedule 3. Decide if she should accept any special orders 4. Decide if she should put least popular Meals on special for a day when demand is low 5. Decide if Available quantity should be increased on some Meals and reduced on others. |

**Report 3 Design**

****

****

**Testing**

The “black box” functional tests are conducted to determine the system’s functionality features and how the system will handle extremities that users may ever possibly come up with*.*

|  |  |
| --- | --- |
| **Idea 1:** | Test that when a customer places an order on a product the system makes the correct decision based on the availability of the product |
| **Justification:** | Our client, because she only prepares Meals as they are ordered, she has limited quantity of Meals she can prepare in a given week, therefore, orders placed cannot exceed the available quantity specified.  This test will test that orders placed do not, under any condition, exceed the available quantity. Ensuring this will also prevent users’ frustration of placing orders only to be told that the Meals they ordered cannot be prepared. |
| **Possible test scenarios:** | * All Meals that appear in the “Available” list have Product\_AvailableQty greater than zero. * System’s response when a user tries to order quantity that is greater than Product\_AvailableQty * System’s response when Product\_AvailableQty = 1 on a certain product and two users concurrently try to place an order that product. (This test will be extended to multiple Meals, e.g. when Product\_AvailableQty = 5 and one user tries to place an order for 3 Meals and another places an order for 5) * When the Available quantity becomes zero the system immediately sets the Product’s status to “unavailable” * When a user places items in the cart and leaves them there, if the product becomes unavailable, the product must be clearly marked for the user to decide if he/she wants to make a special order (special orders are orders on Meals that are not available, the client has to accept or reject these orders depending whether or not she can prepare them) * Meals that are placed in the cart from the “unavailable” list but no order is placed on them and the Meals become “available”, they should be clearly marked. |

|  |  |
| --- | --- |
| **Idea 2:** | Customer’s account and its data are persistent when changes happen |
| **Justification:** | Data persistence is an important part of enhancing usability and reducing the users’ frustrations with losing their data and having to redo tasks.  Customer accounts are important for tracking the customers’ orders for correct order deliveries and collection of payments.  Keeping customer accounts unique is important for ensuring that customer login credentials are not confused and allows login into incorrect accounts. |
| **Possible test scenarios:** | * When a customer creates an account, the email address is validated to ensure it is unique to that customer’s account * When a customer logs in and “accidentally” click the back button he/she can go back into his/her account without having to input login credentials again * Items that have been placed in the cart stay there after the customer logs out and logs in on a different day/browser * Items that have been placed in the cart are not erased when the customer updates account details, e.g. password change does not affect the user’s cart |

|  |  |
| --- | --- |
| **Idea 3:** | Order statuses follow a logical order |
| **Justification:** | Order statuses have to be correctly reflected to enable the administrator and customers to track orders, therefore, the system should ensure that the status changes follow a logical order. |
| **Possible test scenarios:** | * Order statuses cannot do not go back i.e. an order cannot be “ready for delivery/collection” then go back to “Awaiting preparation” * Order status changes to “cancelled” when the delivery/collection date lapses before payment is made. * Administrator can update the order status to “Awaiting preparation” from “Unpaid” without affecting the amount payable by the customer, (this is the case where the customer has negotiated with the client to make a payment at a later stage |

|  |  |
| --- | --- |
| **Idea 4:** | Creation and edition of customer accounts and orders are accurately captured |
| **Justification:** | To ensure that a user is entering relevant and appropriate data, this test needs to be done to ensure the correct capturing of accurate data. If users input “nonsense” or the wrong data, the client will not be able to perform her function of managing her customers and their orders and it will not support reporting functionality on the system. |
| **Possible test scenarios:** | * When customers are creating their accounts, the details entered match the data types specified in the database so there is no conflict (validation should be implemented) * The users are asked to fill in their details that are relevant to the client and her business * When a user changes his/her details, the changes are saved on the system once the user has updated the data and the necessary updates are made to the database * When a user is entering in data and an unexpected event causes the system to temporarily shut down, the users should be able to retrieve lost data * Access management is enforced on the system and there are restrictions imposed to what certain users may do. A customer cannot change product details and update statuses. The client cannot manipulate customer’s orders * When a user archives an entry in the database, all related (or dependent) data is archived with it * Users should not have multiple accounts linked to one email address as user’s login with their email and password. If multiple accounts for one email exist, users may not know which account they are signing into which can be frustrating and also confusing for the administrator * The delivery and collection should not be less than 3 days from the current date |

|  |  |
| --- | --- |
| **Idea 5:** | Process of updating product availability needs to follow a logical order and provide necessary information to allow customers to order |
| **Justification:** | Customers need to be able to see the availability of Meals they want before they order it. It influences their purchase. Therefore the client should be able to update availability of Meals because she prepares Meals as orders come in and the customer needs to know which Meals are on offer and how much is available |
| **Possible test scenarios:** | * When the client updates the Product status to “available”, the customer will be able to see these changes * When the updating the quantity available it has to be the number of single Meals that are available |

**Input Testing**

It is necessary to have input testing in order to check for validations, and input checks can be made. To make sure that the data is reliable and minimise human error

According to all our use cases these checks can be done for form inputs.

This test will ensure that users feed the system with reliable data that is clear of any errors during our flow of activities in the use cases; create order and create account.

Security

A comprehensive security testing framework deals with validation across all layers of the system. The primary objective for testing the security of the system is to identify potential vulnerabilities or security holes and to repair them through testing. This will also test that all required fields are entered and that only authorized personals are the ones who will be able to use the system.

Test Objectives

**Login/ Registration testing:**

It is essential to test registration and login as the user cannot exist on the database therefore will not be able to login without registering, thus the main functionality of the system cannot be met.

These tests will ensure our use cases; create order and create account are functioning to their best capacity ensuring that it improves user experience.

* Checks if email exists
* Checks if Password has at least 6 characters with the use of both uppercase and lower-case letters (case sensitivity), inclusion of one or more numerical digits, inclusion of special characters, such as @,%, $ #, $
* Checks if validation rules are met
* Checks if all required fields are filled in
* Checks if the password is hidden whilst typing it show to enhance privacy
* Checks if the “Remember Me” feature works

**Usability Testing**

It is necessary to have a system that conforms to conventions, to ensure ease of use and improve the user experience

* Check if buttons are clickable
* Check if image has been hyperlinked properly
* Check functionality of button
* Check if button loads on most browsers

**Database Testing**

Database testing involves the process of testing procedures and field size of data to check if the exact values which have been retrieved from the database by the web application correctly match as per the records that are stored in the Database. Therefore performing this testing is important as it will grant the tester any defects that are in the database. Database testing also seeks to ensure the validity and integrity of data stored-if correct input and output is generated. This type of testing also allows one to test the Create, Retrieve, Update and Delete (CRUD) function which ensures that all the data of customers that's supposed to go into the database properly, deleting and updating of tables or records.