AN ELECTRONIC COMMERCE WEBSITE FOR A BOOK SHOP USING EXPRESS.JS SERVER WITH CUSTOMER CHURN PREDICTION USING NEURAL NETWORK

CHAN KHAI SHEN (11323943)

INTRODUCTION

INTRODUCTION (ELECTRONIC COMMERCE)

- The size of electronic commerce sector in Malaysia is around $ 2 billion in 2020 and is projected to be $ 3.6 billion in 2025
- Electronic commerce is based on website or smartphone application
- So, there is a large amount of customer data being collected by companies
- So, customer analytics is possible

INTRODUCTION (CUSTOMER CHURN)

- Churn analysis is part of customer analytics
- Customer churn is a condition where a customer discontinues the product or service of a company
- Customer churn in electronic commerce sector is defined as the customer will not interact with the website in next month
- Customer churn rate in online retail sector is about 21%
- Customer churn will result in significant revenue lost
- Severe customer churn will impact the chance of survival of a company

INTRODUCTION (CHURN PREDICTION)

- Customer acquisition is about 5-25 times more expensive than customer retention
- Customer retention program on the whole customer base is more expensive than targeted customer retention program
- So, the best solution is targeted customer retention on customer who are likely to churn
- Customer churn prediction can find out customers who are likely to churn in the near future

PROBLEM STATEMENT 1

- Sometimes when customer goes to the physical shop, the shop is not opening
- Online shop does not have opening hours, as it is opened 24 hours a day
- Online shop can provide extra business opportunity to the shop
- Online shop can automate the order process, so reducing human error
- So, an electronic commerce book shop website will be developed

PROBLEM STATEMENT 2

- Customer churn can cause significant revenue lost to the online shop
- Customer churn in online retail is around 21%
- To reduce customer churn, customer retention is 5-25 times cheaper than customer acquisition
- Targeted retention is cheaper than retention on the whole customer base
- To identify customer who is at risk of churning, churn prediction is needed
- So, customer churn prediction model will be built on the online book shop

DETAILED RESEARCH QUESTIONS

- How can an electronic commerce website for a book shop be built?
- How can customer churn prediction model for electronic commerce based on neural network be built and integrated into the website?

OBJECTIVES

- To develop an electronic commerce website for a book shop using Express.js, which allows the administrator to create stock and upload item description, and the customer to choose and place order online
- To build a customer churn prediction model for electronic commerce based on neural network and integrate the model into the online book shop website, by displaying the predictions for each customer in a table, to the administrator

CLIENT, AUDIENCE AND MOTIVATION

- The proposed system has two main clients:

  - Book shop customer

  - Book shop owner

CLIENT, AUDIENCE AND MOTIVATION (CONTINUED)

BOOK SHOP CUSTOMER

- Can visit the book shop website at anytime and anywhere, regardless of the opening hours of the physical shop

BOOK SHOP OWNER

- Can view the probability of churning of each customer, gaining insights for arranging strategy for customer retention

SCOPE

- The proposed system is a website for one book shop only
- The proposed system has two types of user:

  - Customer

  - Administrator

SCOPE (CUSTOMER)

- Can view list of items and item details (Login not needed)
- Can register (Login not needed)
- Can add item to cart (Login needed)
- Can view cart and remove item from cart (Login needed)
- Can place order (Login needed)
- Can view list of past orders and order details (Login needed)

SCOPE (ADMINISTRATOR)

- Pre-registered from coding
- Login is needed for all features
- Can upload name, description, weight, price, quantity and one image for each item
- Can edit item details and delete item
- Can view list of orders and order details

SCOPE (CHURN PREDICTION)

- Can save the following predictors for each customer:

  - logout timestamp

  - user-application interaction (view item, add to cart, place order) timestamp

  - transaction timestamp and amount

- Can display the probability of churning for each customer, to the administrator

CONTRIBUTION

- Developed an electronic commerce book shop website which allows the administrator to update stock and view order and allows the customer to browse through the items selling on the website, choose item and place order
- Built a customer churn prediction model for electronic commerce
- Developed functionalities to capture each customer's logout time stamps, user-application interaction (view item, add to cart) time stamps and transaction time stamp and amount and to compute input attributes of the churn prediction model using the aforementioned captured attributes
- Allowed the administrator to view the churn probability of each customer

LITERATURE REVIEW

LITERATURE REVIEW (ELECTRONIC COMMERCE WEBSITE)

- Reviewed definition, contributing factors and advantages of electronic commerce
- Reviewed business-to-consumer electronic commerce
- Reviewed the functionalities and design of existing electronic commerce websites
- Compared the functionalities of reviewed works and the proposed work

LITERATURE REVIEW (CUSTOMER CHURN PREDICTION IN ELECTRONIC COMMERCE USING NEURAL NETWORK)

- Reviewed the existing work of customer churn prediction model for electronic commerce, telecommunication and banking
- Compared the highest accuracy achieved and the algorithm of best accuracy of the reviewed works
- Reviewed the concept of artificial neural network
- Reviewed feed-forward neural network and cross entropy cost function

METHODOLOGY

ITERATIVE MODEL

- In iterative model, in each iteration, one part of the software is specified and implemented and a new version of the software is produced at the end of each iteration
- The proposed system is implemented in 7 iterations
- This will be explained in detail in the Gannt chart in the following slide

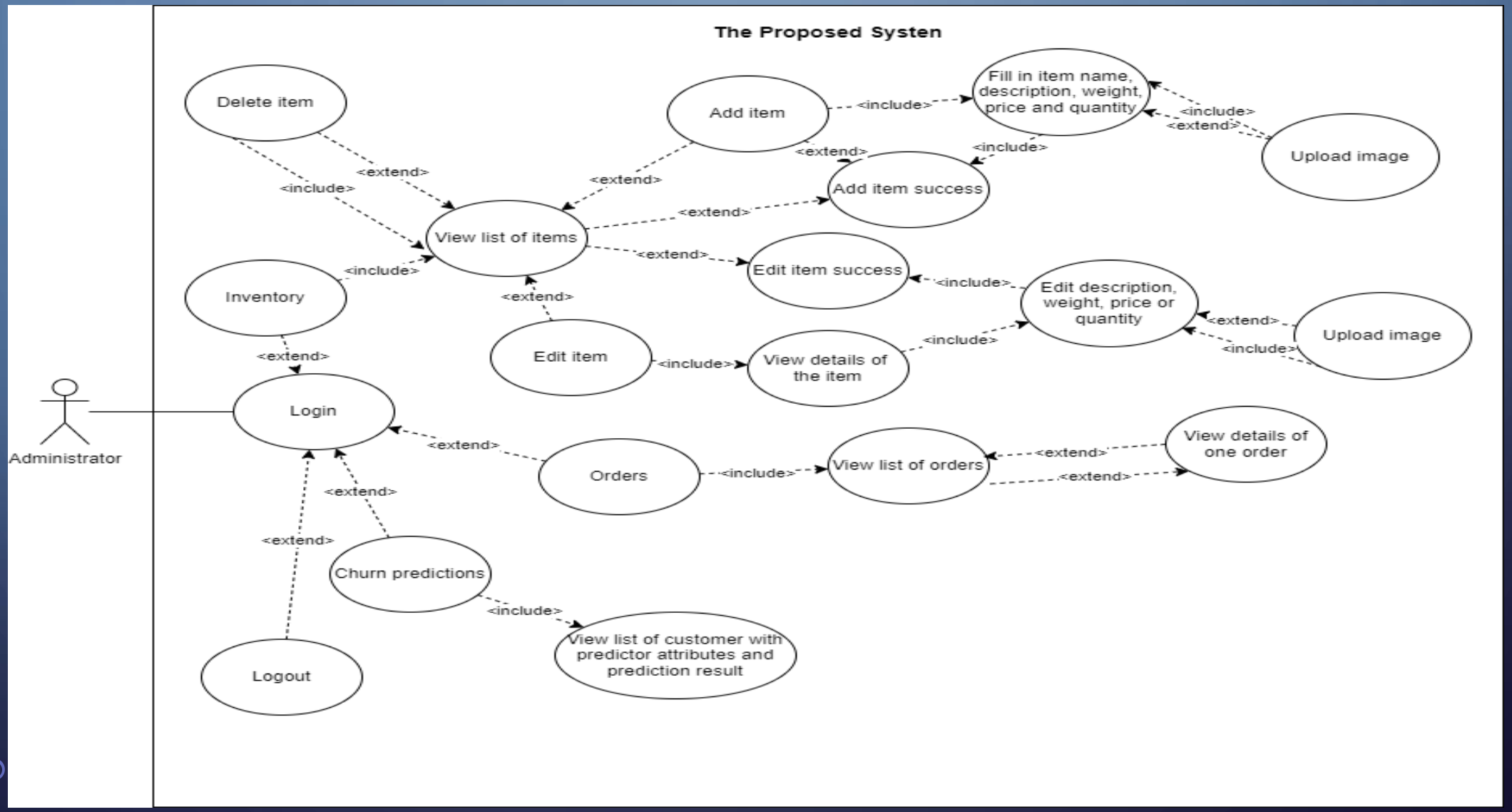
GANNT CHART

Item	18/4 – 2/5	3/5 – 8/5	9/5-15/5	16/5-20/5	21/5-28/5	29/5-4/6	5/6-13/6
Administrator Login	/						
Inventory page	/						
Add Item	/						
Edit Item	/						
Delete Item	/						
Find dataset		/					
Analyse dataset		/					
Dump dataset to database			/				
Train model			/				
Predict dummy data			/				
Churn predictions page				/			
Browse item page					/		
View item details					/		
Customer registration					/		
Add item to cart					/		
View cart						/	
Reduce/increase item quantity						/	
Place order						/	
Orders page							/
View order details							/

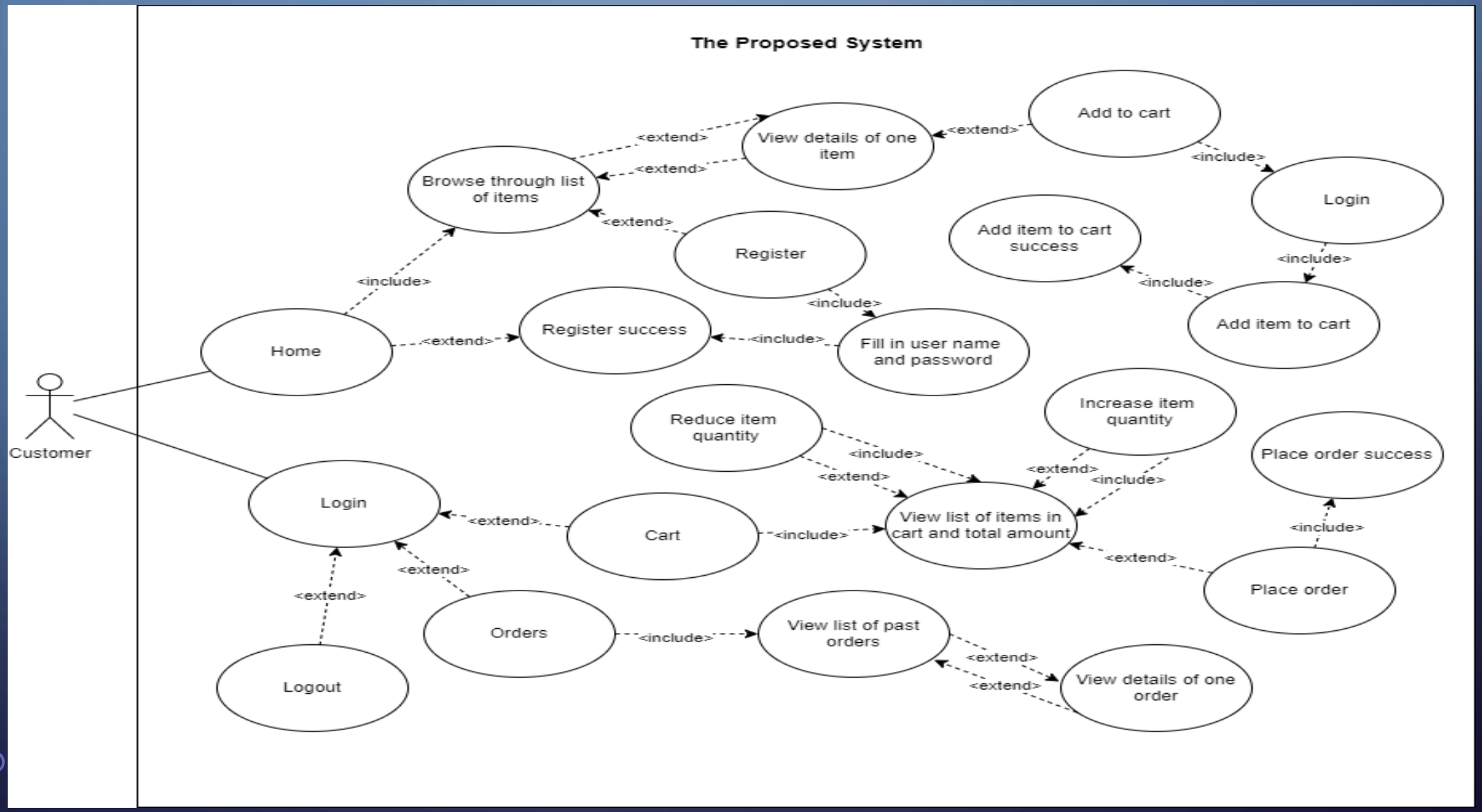
TYPES OF USER

- The proposed system has 2 types of user, namely administrator and customer
- Administrator is the book shop owner
- Customer is the book shop customer
- The use case of each user will be explained in the following slides

USE CASE DIAGRAM (ADMINISTRATOR)



USE CASE DIAGRAM (CUSTOMER)



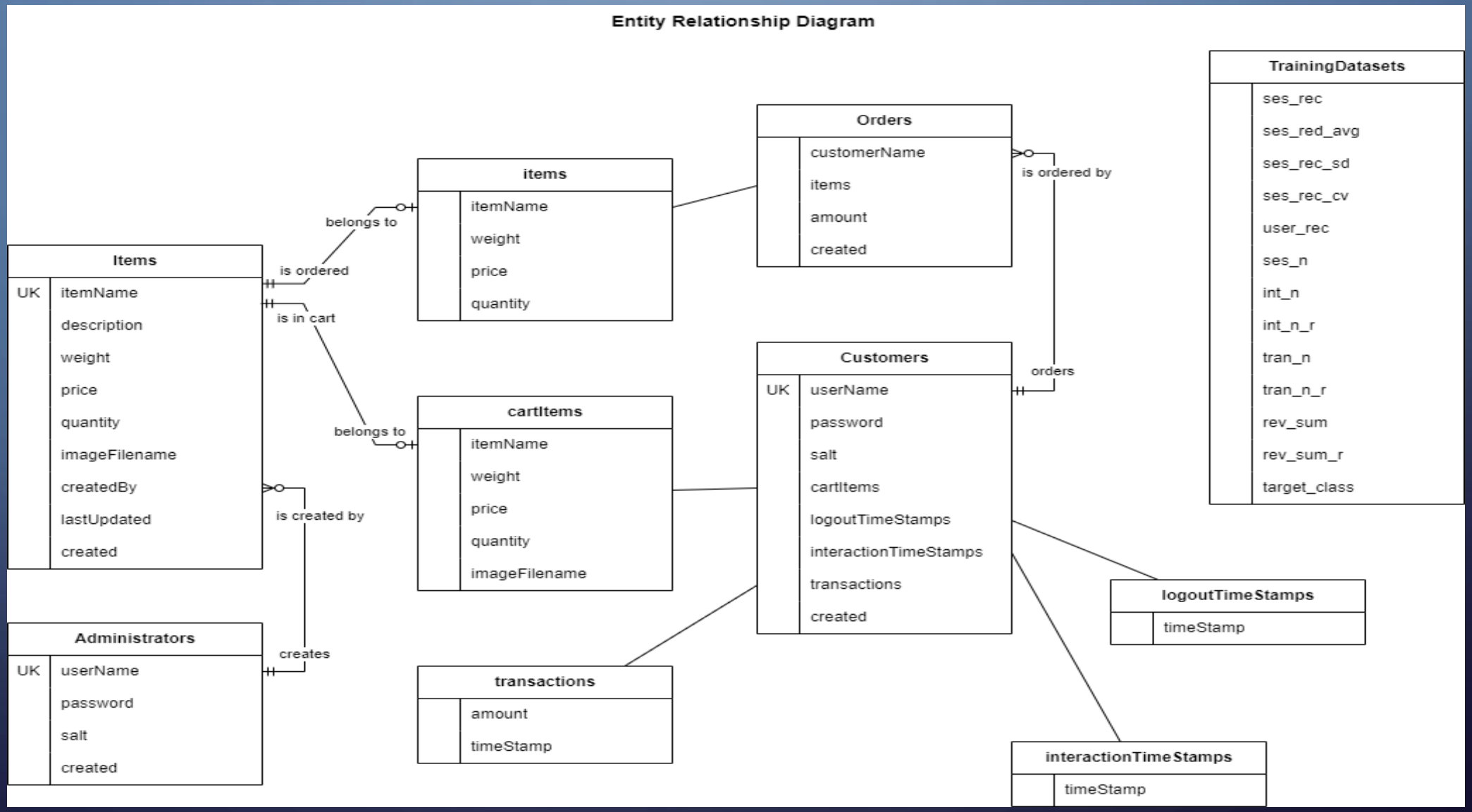
DATABASE ENTITY

- In the proposed work, MongoDB database is used to store the data
- The collections in the database are administrators, items, customers, orders and training datasets
- Training datasets collection holds the training dataset used to train the customer churn prediction model and have no relation to the other collections
- On the other hand, administrators, items, customers and orders collections are related

ENTITY RELATIONSHIP

- Each item is created by only one administrator and each administrator can create zero or many items
- Each order is created by only one customer and each customer can create zero or many orders
- Each item in each order is belonged to only one item in items collection and each item in items collection can exist in one order for zero or one time
- Each cart item of each customer is belonged to only one item in items collection and each item in items collection can exist in the cart of one customer for zero or one time

ENTITY RELATIONSHIP DIAGRAM



CUSTOMERS COLLECTION

Field	Explanation
userName	String, maximum length 20, unique key
password	String
salt	String
cartItems	Array
logoutTimeStamps	Array
interactionTimeStamps	Array
transactions	Array
created	Date

CARTITEMS IN CUSTOMERS COLLECTION

Field	Explanation
itemName	String
price	Number
weight	Number
quantity	Number
imageFilename	String

LOGOUTTIMESTAMPS IN CUSTOMERS COLLECTION

Field	Explanation
timeStamp	Date

This information is used for churn prediction and will be explained in detail later.

INTERACTIONTIMESTAMPS IN CUSTOMERS COLLECTION

Field	Explanation
timeStamp	Date

This information is used for churn prediction and will be explained in detail later.

TRANSACTIONS IN CUSTOMERS COLLECTION

Field	Explanation
amount	Number
timeStamp	Date

This information is used for churn prediction and will be explained in detail later.

ITEMS COLLECTION

Field	Explanation
itemName	String, maximum length 80, unique key
description	String
weight	Number
price	Number
quantity	Number
created	Date

ADMINISTRATORS COLLECTION

Field	Explanation
userName	String, maximum length 20, unique key
password	String
salt	String
created	Date

ORDERS COLLECTION

Field	Explanation
customerName	String
items	Array
amount	Number
created	Date

ITEMS IN ORDERS COLLECTION

Field	Explanation
itemName	String
price	Number
weight	Number
quantity	Number

TRAINING DATASETS COLLECTION

Field	Explanation
ses_rec	Number
ses_rec_avg	Number
ses_rec_sd	Number
ses_rec_cv	Number
user_rec	Number
ses_n	Number
int_n	Number
int_n_r	Number
tran_n	Number
tran_n_r	Number
rev_sum	Number
rev_sum_r	Number
target_class	Number

THE TRAINING DATASET

- The training dataset is originated from the work of Fridrich & Dostal (2022)
- It is available on Kaggle from https://www.kaggle.com/datasets/fridrichmrtn/user-churn-dataset
- The training dataset has one predicted variable which is churn or not churn
- The training dataset has multiple predictor variables which is generally available for electronic commerce retail store
- 12 predictor variables are selected to be used in the proposed system
- Selection is made based on the ability of the proposed system to capture the predictors

PREDICTOR VARIABLES (PART 1 OF 3)

Attribute name	Variable name	Explanation	Ways of getting the variable in the proposed system
Session recency	ses_rec	Difference in days between the last user session and the second last user session in the past month	$Date\ of\ last\ logout - Date\ of\ second\ last\ logout$
Average session recency	ses_rec_avg	Average days between sessions	Refer to code
Standard deviation in session recency	ses_rec_sd	Standard deviation of days between sessions	Refer to code
Coefficient of variation in session recency	ses_rec_cv	Ratio of standard deviation in session recency to average session recency	$\frac{Standard\ deviation\ in\ session\ recency}{Average\ session\ recency}$

PREDICTOR VARIABLES (PART 2 OF 3)

Attribute name	Variable name	Explanation	Ways of getting the variable in the proposed system
User maturity	user_rec	Difference in days between the last user session and the first user session	$Date\ of\ last\ logout - Date\ of\ first\ logout$
Session frequency	ses_n	Number of sessions	$Number\ of\ logouts$
User-app interaction frequency	int_n	Number of user-application interaction (view item / add to cart / place order)	$Number\ of view\ and\ add\ to\ cart + Number\ of\ transaction$
Relative user-app interaction frequency	int_n_r	Ratio of user-app interaction frequency to session frequency	$\frac{User\ app\ interaction\ frequency}{Session\ frequency}$

PREDICTOR VARIABLES (PART 3 OF 3)

Attribute name	Variable name	Explanation	Ways of getting the variable in the proposed system
Transactional frequency	tran_n	Number of transaction	$Number\ of\ transaction$
Relative transactional frequency	tran_n_r	Ratio of transactional frequency to session frequency	$\frac{Transactional\ frequency}{Session\ frequency}$
Transactional revenue	rev_sum	Total revenue	$Sum\ of\ transaction\ amount$
Relative transactional revenue	rev_sum_r	Ratio of transactional revenue to session frequency	$\frac{Transactional\ revenue}{Session\ frequency}$

DUMPING OF TRAINING DATASET INTO THE DATABASE

- The training dataset, which is in comma separated values (.csv) format, is parsed using “fs” library and “csv-parse” library and changed to JavaScript object before it is saved into the MongoDB database
- The training dataset is retrieved from the database and changed to array before it is passed as the training dataset of the customer churn prediction model

TRAINING THE CHURN PREDICTION MODEL USING NEURAL NETWORK

- An experiment conducted by Ismail, et. al. (2015) found that multi-layer perceptron (neural network) has better performance compared to logistic regression and multiple regression
- So, in the proposed system the customer churn prediction model is trained using neural network
- TensorFlow.js library is used for training the customer churn prediction model

CONFIGURATION OF THE TRAINING OF THE CHURN PREDICTION MODEL (PART 1 OF 2)

- An experiment conducted by Pondel, et. al. (2021) found that the test accuracy of 1 dense hidden layer without dropout with Rectified Linear Unit (ReLU) activation function is higher (0.736) than that of 2 dense hidden layers with 30% dropout with Leaky ReLU activation function (0.733)
- So, in the proposed system, the customer churn prediction model is trained with 1 dense input layer with 4 neurons with ReLU activation function, 1 dense hidden layer with 4 neurons with ReLU activation function and 1 dense output layer with 1 neuron with sigmoid activation function

CONFIGURATION OF THE TRAINING OF THE CHURN PREDICTION MODEL (PART 2 OF 2)

- An experiment conducted by Domingos, et. al. (2021) has found that Root Mean Square Propagation (RMSProp) optimizer achieved best accuracy for multi-layer perceptron (dense neural network)
- So, in the proposed system, during training of the churn prediction model, RMSProp optimizer is used
- The other configurations are binary cross entropy loss function, batch size 100, number of epochs 200 and validation split 30%

PRE-TRAINED CHURN PREDICTION MODEL

- The churn prediction model is pre-trained on the server using the configurations mentioned in the previous slides
- The model is then saved on the server
- Every time when the administrator accesses the churn predictions page, the pre-trained prediction model is loaded to carry out churn prediction on each customer

TRAINING AND SAVING CUSTOMER CHURN PREDICTION MODEL (CODE)

```
const model = tf.sequential({
    layers: [
        tf.layers.dense({inputShape: [12], units: 4, activation: 'relu'}),
        tf.layers.dense({units: 4, activation: 'relu'}),
        tf.layers.dense({units: 1, activation: 'sigmoid'})
    ]
});

model.compile({
    optimizer: 'rmsprop',
    loss: 'binaryCrossentropy',
    metrics: ['accuracy']
});

model.fit(x_tensor, y_tensor, {
    batchSize: 100,
    epochs: 200,
    validationSplit: 0.3
}).then(info=>{
    console.log('Train arccuracy', info.history.acc[199]);
    console.log('Validation accuracy', info.history.val_acc[199]);

    const prediction = model.predict(tf.randomNormal([5, 12]));
    prediction.print();

    // Save the model

    model.save('file://' + path.join(__dirname + '/ChurnModel')).then(response=>{
        console.log('Saved', response);
```

GET PREDICTORS EXPLANATION

- Every time when the customer logouts, the time stamp will be recorded (analogous to user session)
- Every time when the customer views the details of an item and add an item to cart, the time stamp will be recorded (analogous to user-application interaction)
- Every time when the customer places order, the time stamp and the amount will be recorded (analogous to transaction)
- These 3 pieces of information will be computed into the 12 predictor variables (code in following slides)

GET PREDICTORS CODE (PART 1 OF 3)

```
// get predictor values
function getPredictors(logoutTimeStamps, viewAndAddToCartTimeStamps, transactions) {

    logoutTimeStamps.sort((a, b)=>{return new Date(a.timeStamp).getTime() - new Date(b.timeStamp).getTime();});

    //console.log(logoutTimeStamps);

    var ses_rec;
    if (logoutTimeStamps.length < 2) {
        ses_rec = 31;
    }
    else {
        //console.log(logoutTimeStamps[logoutTimeStamps.length - 2].timeStamp, new Date(logoutTimeStamps[logoutTimeStamps.length - 1].timeStamp).getTime())

        const differenceInMilliseconds = new Date(logoutTimeStamps[logoutTimeStamps.length - 1].timeStamp).getTime() - new Date(logoutTimeStamps[logoutTimeStamps.length - 2].timeStamp).getTime();
        const differenceInDays = differenceInMilliseconds / (1000 * 3600 * 24);

        if (differenceInDays > 31) {
            ses_rec = 31;
        }
        else {
            ses_rec = differenceInDays;
        }
    }
```

GET PREDICTORS CODE (PART 2 OF 3)

```
const listOfDayBetweenSessions = [];
if (logoutTimeStamps.length < 2) {
    listOfDayBetweenSessions.push(99);
}
else {
    for (var j = 0; j < logoutTimeStamps.length - 1; j++) {
        const differenceInMilliseconds = new Date(logoutTimeStamps[j + 1].timeStamp).getTime() - new Date(logoutTimeStamps[j].timeStamp).getTime();
        const differenceInDays = differenceInMilliseconds / (1000 * 3600 * 24);

        if (differenceInDays > 99) {
            listOfDayBetweenSessions.push(99);
        }
        else {
            listOfDayBetweenSessions.push(differenceInDays);
        }
    }
}

const ses_rec_avg = mathjs.mean(listOfDayBetweenSessions);

const ses_rec_sd = mathjs.std(listOfDayBetweenSessions);

const ses_rec_cv = ses_rec_sd / ses_rec_avg;
```

GET PREDICTORS CODE (PART 3 OF 3)

```
var user_rec;
if (logoutTimeStamps.length < 2) {
    user_rec = 99;
}
else {
    const differenceInMilliseconds = new Date(logoutTimeStamps[logoutTimeStamps.length - 1].timeStamp).getTime() - new Date(logoutTimeStamps[0].timeStamp).getTime();
    const differenceInDays = differenceInMilliseconds / (1000 * 3600 * 24);

    if (differenceInDays > 99) {
        user_rec = 99;
    }
    else {
        user_rec = differenceInDays;
    }
}

const ses_n = logoutTimeStamps.length == 0 ? 1 : logoutTimeStamps.length;

const int_n = viewAndAddToCartTimeStamps.length + transactions.length;

const int_n_r = int_n / ses_n;

const tran_n = transactions.length;

const tran_n_r = tran_n / ses_n;

var rev_sum = 0;
for (var i = 0; i < transactions.length; i++) {
    rev_sum = rev_sum + transactions[i].amount;
}

const rev_sum_r = rev_sum / ses_n;

return [ses_rec, ses_rec_avg, ses_rec_sd, ses_rec_cv, user_rec, ses_n, int_n, int_n_r, tran_n, tran_n_r, rev_sum, rev_sum_r];
```

CHURN PREDICTIONS PAGE

- The churn predictions page is only available to the administrator
- When the churn predictions page is loaded, the logout time stamps, interaction time stamps and transactions of each customer is retrieved from the database
- These 3 pieces of information are computed into the 12 predictor variables
- These 12 predictor variables are passed as the input of the customer churn prediction model
- Finally, the 12 predictor variables and the prediction result (probability of churn) of each customer are displayed in a table on the web page

BASIC AUTHENTICATION LOGIN

- In the proposed system, basic authentication login is used
- When login is required, the server responds with error code “401” and “WWW-Authenticate” header as “Basic” to trigger the basic authentication process
- When the server receives the response from the client, the server retrieves the user name and password from the “Authorization” header

PASSWORD HASHING USING PBKDF2

- Upon registration, the password is hashed using Password-Based Key Derivation Function 2 (PBKDF2) before it is stored in MongoDB database
- Upon login, the input password is hashed with the stored salt and compared with the stored hashed password
- If the hashed input password is same as the stored hashed password, the user (customer / administrator) is given access

GENERAL PROCESS OF ELECTRONIC COMMERCE WEBSITE

Upload Item → View Item → Add to Cart → Check out for payment

Note: In the proposed system, there is no payment feature, but there is a place order feature. The payment feature is placed as future enhancement. The customer can make payment and collect the ordered item at the physical shop.

INVENTORY PAGE

- The inventory page is only available to the administrator
- On the inventory page, the items selling on the proposed system are listed in a table
- The image, name, unit weight, unit price, stock quantity and last updated date of each item are displayed
- From the inventory page, the administrator can access the create, read, update and delete (CRUD) functionalities

ADD NEW ITEM PAGE

- From the inventory page, the administrator can click on the “Add New Item” button to go to the add new item page
- The add new item page is only available to the administrator
- On the add new item page, the administrator can key in name, description, unit weight, unit price and stock quantity for the item
- Optionally, the administrator can upload one image for the item
- The add new item page is rendered using HTML Form

HANDLING IMAGE FILE

- The image is sent through post method and “multipart/formdata” encryption
- On Express.js server, the image is handled by using the “multer” library
- The image is saved on the server’s “upload” directory with the file name of <time stamp>-<original file name>
- Only the image file name is saved on the MongoDB database
- The image file name is unique, so it can be used to access the image file in future

EDIT ITEM PAGE

- From the inventory page, the administrator can edit an item by clicking on the "Edit" button on the same row as that item
- The edit item page is only available to the administrator
- On the edit item page, name, description, unit weight, unit price, stock quantity and image of that item are displayed
- The administrator can edit every field except the item name
- The administrator can also change the image of the item
- The edit item page is rendered as HTML Form

DELETE ITEM

- On the inventory page, the administrator can delete an item by clicking on the "Delete" button on the same row as that item
- Once the item is deleted, the page is reloaded to render the updated list of items

HOME PAGE

- The home page is available to any type of user, but it is meant for the customer
- On the home page, all items selling on the proposed system are displayed in grid arrangement
- The image, name and price of each item are displayed
- On the home page, the customer can browse through all the items
- From the home page, the customer can opt to login, register or view the item details

HOME PAGE (LOGGED IN VS NOT LOGGED IN)

- If the customer is not logged in, “Login” button and “Register” button will appear on the top navigation bar
- Else if the customer is logged in, “Logout” button will appear on the top navigation bar
- After the home page is loaded, a get request will be sent to the server, and the server will respond in text either “Logged in” or “Not logged in”, depending on whether there is an “Authorization” header or not
- The home page will render differently according to the response

CUSTOMER REGISTRATION PAGE

- From the home page, the customer can click on the “Register” button to go to the customer registration page
- The customer registration page is available to any type of user, but it is meant for the customer
- The customer needs to key in user name and password to register
- The customer needs to repeat password to complete registration

ITEM DETAILS PAGE

- From the home page, the customer can view the details of an item by clicking on the “Details” button of that item
- The item details page is available to any type of user, but it is meant for the customer
- On the item details page, image, name, unit weight, unit price and description of that item are displayed
- On the item details page, the customer can add the item to cart (details on next slide)

ADD TO CART

- On the item details page, when the customer click on the “Add To Cart” button, the item will be added to the cart
- If the customer is not logged in, the customer will be prompted to login
- If the item is already in the cart, then the quantity of the item will be increased by 1

CART PAGE

- The cart page is only available to the customer
- The cart page provides a summary of the items that are currently in the cart
- On the cart page, the items in the cart are listed in table
- On the cart page, image, name, unit weight, unit price, quantity and total price of each item are listed
- On the cart page, the amount (price) of the items in the cart is displayed
- The customer can increase or decrease the quantity of an item in the cart (details in next slide)
- On the cart page, the customer can place order (details in following slide)

INCREASE / DECREASE ITEM QUANTITY IN CART

- On the cart page, the customer can increase the quantity of an item by clicking on the “Add One” button on the same row as that item
- The customer can also decrease the quantity of an item by clicking on the “Minus One” button on the same row as that item
- If the quantity after decreased is equal to 0, then the cart page will be reloaded to render the updated items in the cart

PLACE ORDER

- On the cart page, the customer can click on the “Place Order” button to place order
- First, the server checks whether the stock quantity is enough
- If the stock quantity is enough, the stock quantity in the database is decreased accordingly
- Then, all items in the cart are deleted in the database
- Finally, the order is saved in the database

ORDERS PAGE

- The orders page is available to the administrator and the customer (both need login)
- On customer side, only the orders placed by that customer are listed
- On administrator side, all orders placed in the proposed system are listed
- The orders are listed in a table
- ID, date and amount of each order are displayed
- On administrator side, the customer name of each order is displayed

ORDER DETAILS PAGE

- On the orders page, the user (customer / administrator) can view the details of an order by clicking on the “Details” button on the same row as that order
- On the order details page, all items in the order are listed in table
- On the order details page, name, unit weight, unit price, quantity and total price of each item are listed
- On the order details page, the total amount of the order is displayed

EVALUATION / RESULTS

UNIT TESTING (GET PREDICTORS FUNCTION)

- The get predictors function computes logout time stamps, interaction time stamps, transaction time stamps and amounts into the 12 predictor variables
- Unit testing is used to test the get predictors function
- The test cases are formed by inputting different combinations of logout time stamps
- Observation is made by observing the results of ses_rec (difference between last and second last session), ses_rec_avg (average days between session), user_rec (difference between last and first session) and ses_n (number of sessions)

UNIT TESTING (ERROR HANDLING)

- Unit testing is also used to test the error handling of each page
- The pages that are tested are customer registration page, add new item page, edit item page, cart page and order details page
- The test cases are formed by providing input that can trigger error message
- The actual error message is observed and compared with the expected error message

BLACK BOX TESTING (SAVE PREDICTORS FUNCTION)

- The save predictors function records the logout time stamp when the customer logouts, record the interaction time stamp when the customer views item or add item to cart and record the transaction time stamp and amount when the customer places order
- Black box testing is used to test the save predictors function
- The test cases are formed by doing the actions that can trigger the save predictors function as mentioned above

BLACK BOX TESTING (SAVE PREDICTORS FUNCTION) - CONTINUED

- When loading the churn predictions page, the saved attributes (as mentioned in previous slide) will be converted to the 12 predictor variables
- The churn predictions page is loaded before and after the test to observe the change in the value of the predictor variables
- The predictor variables that are observed are ses_n (number of sessions), int_n (number of user-application interactions), tran_n (number of transactions) and rev_sum (total revenue)

BLACK BOX TESTING (UPDATE STOCK FUNCTION)

- Update stock function updates the stock quantity when the customer places order
- Black box testing is used to test the update stock function
- The test case is use a dummy customer account to place order on certain items
- Then, the actual change in stock quantity in the proposed system is observed and compared with the expected change in stock quantity

NON-FUNCTIONAL TESTING (CUSTOMER CHURN PREDICTION MODEL)

- The customer churn prediction model is tested to obtain the number of true positives, the number of false positives, the number of true negatives and the number of false negatives
- The threshold of the test is set at 0.5, which means if the predicted churn probability is larger or equal than 0.5, the result is considered positive (churn), else the result is considered negative (non-churn)
- From the test results, the accuracy of the model is calculated and the confusion matrix is drawn

CONFUSION MATRIX

Actual \ Predicted	Non-churn	Churn
Non-churn	922	4725
Churn	2122	41589

ACCURACY

- There are total of 49358 rows
- The number of true positive, the number of true negative, the number of false positive and the number of false negative are 41589, 922, 4725 and 2122 respectively
- The following formula is used to calculate accuracy:
- $Accuracy = \frac{True\ positive + True\ negative}{True\ positive + False\ positive + True\ negative + False\ negative}$
- The accuracy of the customer churn prediction model is 0.861 or 86.1%

DISCUSSIONS

DISCUSSIONS

- In literature review, it is found that a general electronic commerce website comprises of upload item, view item, add to cart and checkout for payment
- The proposed system comprises of upload item, view item and add to cart
- The proposed system allows customer to place order online and pay and collect item at the physical shop
- In future enhancement, the online payment mechanism will be focused so that customer does not need to go to the physical shop
- In the proposed system, a customer churn prediction model is trained using neural network
- The proposed system is able to capture user session, user-application interaction and transaction information of each customer and transform into the input of the churn prediction model
- The proposed system displays the churn probability of each customer to the administrator in a table

DISCUSSIONS (CUSTOMER CHURN PREDICTION) - CONTINUED

- The accuracy of the customer churn prediction model is 0.861
- This accuracy is not too high and not too low, compared to the accuracies of the reviewed works, which are ranged between 0.740 and 0.976
- In future enhancement, the configuration during training of the customer churn prediction model should be fine tuned so that a higher accuracy can be achieved

CONCLUSION

CONCLUSION

- In the proposed system, the administrator can update stock and view orders
- In the proposed system, the customer can browse through all items, choose items and place order
- In the proposed system, a customer churn prediction model is trained using artificial neural network, achieving accuracy of 0.861
- In the proposed system, the churn probability of each customer can be predicted and displayed to the administrator in a table
- So, both objectives are achieved

THE END

THANK YOU