SYSTEM ANALYSIS AND DESIGN

Deliverable Tasks 2

TASK 1: DATABASE DESIGN

Kindly note that my ERD tables already show multiplicities by the type of relationship line drawn which is, in terms of the UML 2.5, syntactically different from that of the Class Diagrams.

Resolution of Multiple Association: I resolved the many-to-many association that exists between the Admin Class and The Process Management Class by creating an Associative Table.

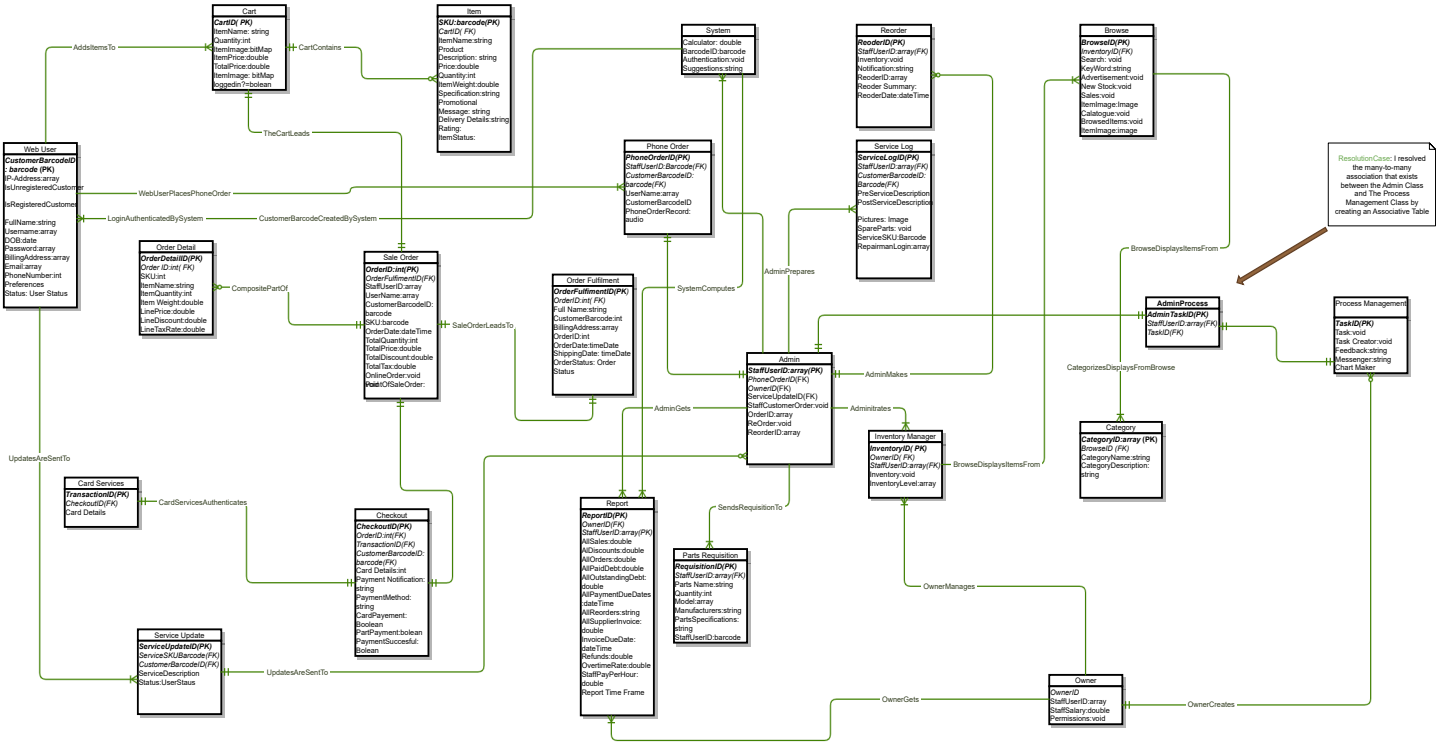
ID Creation: I created new IDs for classes that didn't carry an ID attribute in the Class Diagrams but proves necessary for Data Design purposes. IDs were created for the following classes: Report Class; Parts Requisition, Order Fulfilment, Checkout, Card Services, Order Details, Inventory Manager, Process Management, Owner Id and Browser ID. Noteworthy is the fact that the IDs are specific not just to the type of product or service but the instance of occurrence. For example, the SKU barcode is a unique to an individual item.

Mapping the Web-User Generalization: You will observe the use of the web user generalization at the Class Diagram, with Unregistered User and Register User Subclasses. This I mapped into a single ERD Table. Consequently, The Web-User Superclass is flattened into a single table bearing all the attributes of the Registered and Unregistered subclasses.

Mapping One-to-One Associations: If you observe the Entity Relationship Diagram, you would notice that I cross-inputted the Primary Key of each side. Therefore, each side of the table now has as Functional Key that is a Primary Key of the other table. The Tables that show this are: Card Service---Checkout, Sale Order---Checkout, Phone Order—Admin, Process Management—Admin Process, and Admin—AdminProcess.

Mapping One-to-Many Associations: In my ERD tables are ensured that I transferred the Primary Key of the One-side of the relationship to the Many-side entity. The Entities that reflect this mapping includes: Admin—Parts Requisition, Web User—Phone Order, Admin—Service Log, Owner—Process Management, Sale Order—Order Detail, Admin—Reorder, and Browse—Category.

ERD TABLES FOR THOMPSON CYCLES


Cart
CartID(PK)
ItemName: string
Quantity:int
ItemImage:bitMap
ItemPrice:double
TotalPrice:double
ItemImage: bitMap
loggedin?=bolean
AddsItemsTo
CartContains
TheCartLeads
Item
SKU:barcode(PK)
CartID(FK)
ItemName:string
Product
Description: string
Price:double
Quantity:int
ItemWeight:double
Specification:string
Promotional
Message: string
Delivery Details:string
Rating:
ItemStatus:
System
Calculator: double
BarcodeID:barcode
Authentication:void
Suggestions:string
Reorder
ReoderID(PK)
StaffUserID:array(FK)
Inventory:void
Notification:string
ReorderID:array
Reoder Summary:
ReoderDate:dateTime
Browse
BrowseID(PK)
InventoryID(FK)
Search: void
KeyWord:string
Advertisement:void
New Stock:void
Sales:void
ItemImage:Image
Catalogue:void
BrowsedItems:void
ItemImage:image
Web User
CustomerBarcodeID : barcode (PK)
IP-Address:array
isUnregisteredCustomer
isRegisteredCustomer
FullName:string
Username:array
DOB:date
Password:array
BillingAddress:array
Email:array
PhoneNumber:int
Preferences
Status: User Status
WebUserPlacesPhoneOrder
LoginAuthenticatedBySystem
CustomerBarcodeCreatedBySystem
Phone Order
PhoneOrderID(PK)
StaffUserID:Barcode(FK)
CustomerBarcodeID:
barcode(FK)
UserName:array
CustomerBarcodeID
PhoneOrderRecord:
audio
Service Log
ServiceLogID(PK)
StaffUserID:array(FK)
CustomerBarcodeID:
Barcode(FK)
PreServiceDescription:
PostServiceDescription:
Pictures: Image
SpareParts: void
ServiceSKU:Barcode
RepairmanLogin:array
ResolutionCase: I resolved the many-to-many association that exists between the Admin Class and The Process Management Class by creating an Associative Table
Order Detail
OrderDetailID(PK)
Order ID:int(FK)
SKU:int
ItemName:string
ItemQuantity:int
Item Weight:double
LinePrice:double
LineDiscount:double
LineTaxRate:double
CompositePartOf
Sale Order
OrderID:int(PK)
OrderFulfimentID(FK)
StaffUserID:array
UserName:array
CustomerBarcodeID:
barcode
SKU:barcode
OrderDate:dateTime
TotalQuantity:int
TotalPrice:double
TotalDiscount:double
TotalTax:double
OnlineOrder:void
PastOfSaleOrder:
SaleOrderLeadsTo
Order Fulfilment
OrderFulfimentID(PK)
OrderID:int(FK)
Full Name:string
CustomerBarcode:int
BillingAddress:array
OrderID:int
OrderDate:timeDate
ShippingDate: timeDate
OrderStatus: Order
Status
SystemComputes
AdminPrepares
BrowseDisplaysItemsFrom
AdminProcess
AdminTaskID(PK)
StaffUserID:array(FK)
TaskID(FK)
Process Management
TaskID(PK)
Task: void
Task Creator:void
Feedback:string
Messenger:string
Chart Maker
Admin
StaffUserID:array(PK)
PhoneOrderID(FK)
OwnerID(FK)
ServiceUpdateID(FK)
StaffCustomerOrder:void
OrderID:array
ReOrder:void
ReorderID:array
AdminMakes
AdminGets
Administrates
CategorizesDisplaysFromBrowse
Inventory Manager
InventoryID(PK)
OwnerID(FK)
StaffUserID:array(FK)
Inventory:void
InventoryLevel:array
BrowseDisplaysItemsFrom
Category
CategoryID:array (PK)
BrowseID (FK)
CategoryName:string
CategoryDescription:
string
UpdatesAreSentTo
Card Services
TransactionID(PK)
CheckoutID(FK)
Card Details
CardServicesAuthenticates
Checkout
CheckoutID(PK)
OrderID:int(FK)
TransactionID(FK)
CustomerBarcodeID:
barcode(FK)
Card Details:int
Payment Notification:
string
PaymentMethod:
string
CardPayement:
Boolean
PartPayment:bolean
PaymentSucceful:
Bolean
Report
ReportID(PK)
OwnerID(FK)
StaffUserID:array(PK)
AllSales:double
AllDiscounts:double
AllOrders:double
AllPaidDebt:double
AllOutstandingDebt:
double
AllPaymentDueDates
:dateTime
AllReorders:string
AllSupplierInvoice:
double
InvoiceDueDate:
dateTime
Refunds:double
OvertimeRate:double
StaffPayPerHour:
double
Report Time Frame
SendsRequisitionTo
Parts Requisition
RequisitionID(PK)
StaffUserID:array(FK)
Parts Name:string
Quantity:int
Model:array
Manufacturers:string
PartsSpecifications:
string
StaffUserID:barcode
OwnerManages
Service Update
ServiceUpdateID(PK)
ServiceSKUBarcode(FK)
CustomerBarcodeID(FK)
ServiceDescription
Status:UserStaus
UpdatesAreSentTo
Owner
OwnerID
StaffUserID:array
StaffSalary:double
Permissions:void
OwnerGets
OwnerCreates

TASK 2: DATA NORMALIZATION TABLES

①. ***Q:*** Is there any multivalued attribute?
A: No

Q: Is there any multiple PKs?
A: No
Q: Is there any partial dependency of an Attribute on the PK?
A: No

Q: Is there any transient dependency of A non-key attribute on another non-key Attribute?
A: No
Result: Table is in 3NF

Web User
CustomerBarcodeID: barcode (PK)
IP-Address:array
IsUnregisteredCustomer
IsRegisteredCustomer
FullName:string
Username:array
DOB:date
Password:array
BillingAddress:array
Email:array
PhoneNumber:int
Status: User Status

②. ***Q:*** Is there any multivalued attribute?

A: No (Note that the SKU barcode is Item-specific (not product-specific). 1NF

Q: Is there any multiple PKs?
A: No
Q: Is there any partial dependency of an Attribute on the PK?
A: No. 2NF

Q: Is there any transient dependency of A non-key attribute on another non-key Attribute?
A: Yes
Why? The User Name is dependent on the CustomerBarcodeID

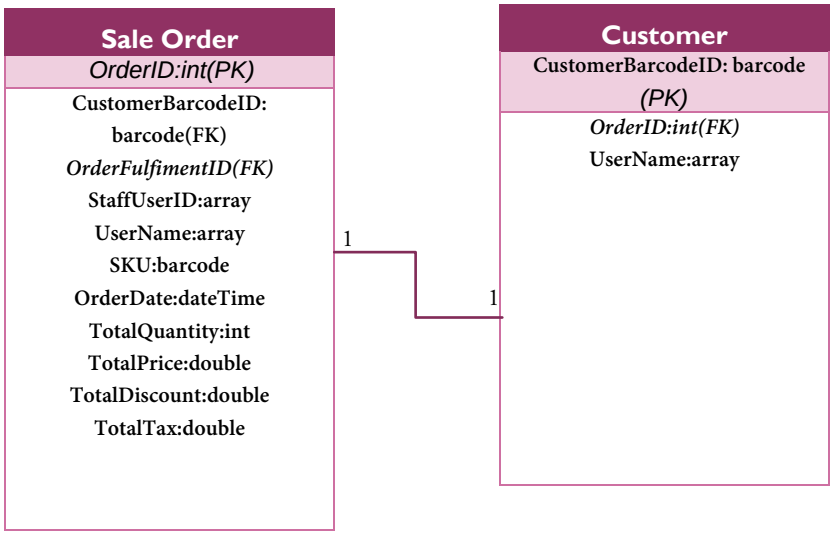
Sale Order
OrderID:int(PK)
OrderFulfimentID(FK)
StaffUserID:array
UserName:array
CustomerBarcodeID: barcode
SKU:barcode
OrderDate:dateTime
TotalQuantity:int
TotalPrice:double
TotalDiscount:double
TotalTax:double

DATA NORMALIZATION TABLES

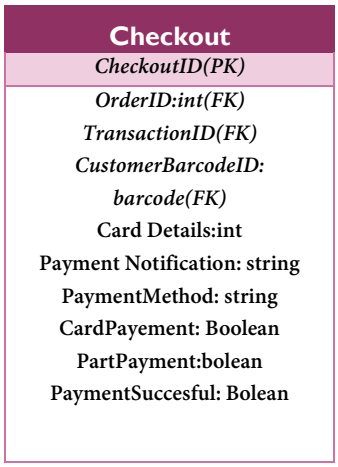
Q: Is there any multivalued attribute?
A: No (Note that the SKU barcode is Item-specific (not product-specific).

Q: Is there any multiple PKs?
A: No
Q: Is there any partial dependency of an Attribute on the PK?
A: No

Q: Is there any transient dependency of A non-key attribute on another non-key Attribute?
A: No
Result: Table is in 3NF



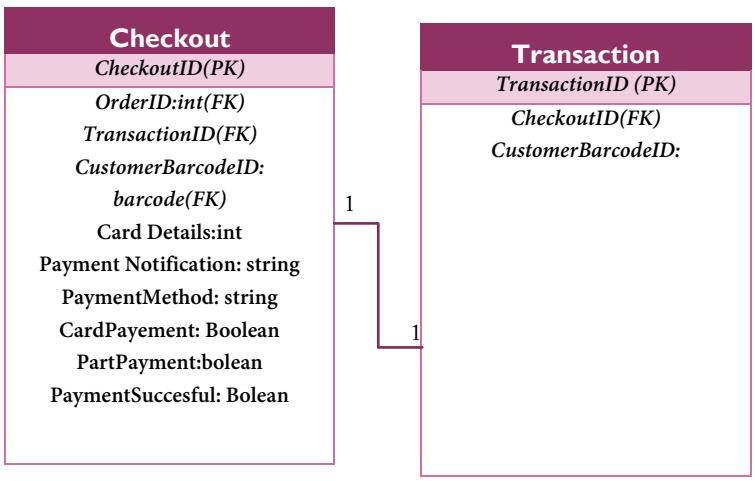
③ ***Q:*** Is there any multivalued attribute?
A: Yes.
Why? In case of failed transactions, a checkout Might have more than one transaction ID



Q: Is there any multiple PKs?

A: No
Q: Is there any partial dependency of an Attribute on the PK?
A: No

Q: Is there any transient dependency of A non-key attribute on another non-key Attribute?
A: No
Result: Table is in 3NF

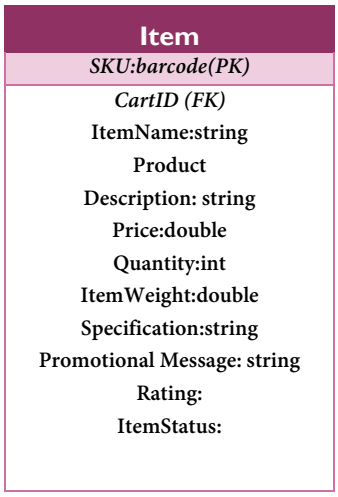


DATA NORMALIZATION TABLES

④ ***Q:*** Is there any multivalued attribute?
A: No

Q: Is there any multiple PKs?
A: No
Q: Is there any partial dependency of an Attribute on the PK?
A: No

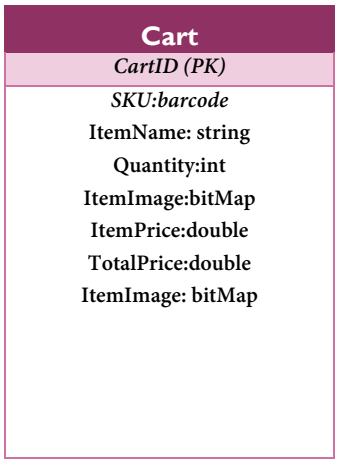
Q: Is there any transient dependency of A non-key attribute on another non-key Attribute?
A: No
Result: Table is in 3NF



⑤. ***Q:*** Is there any multivalued attribute?
A: No

Q: Is there any multiple PKs?
A: No
Q: Is there any partial dependency of an Attribute on the PK?
A: No

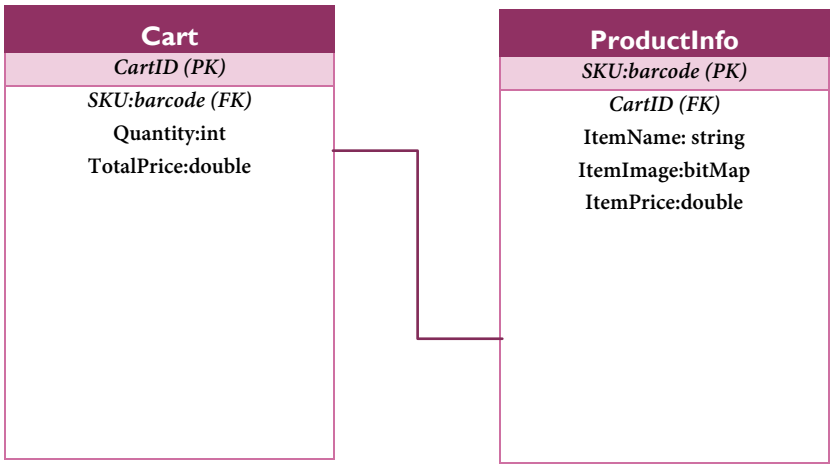
Q: Is there any transient dependency of A non-key attribute on another non-key Attribute?
A: Yes
Why?
The Item Image, Item Price, and Item Name Are dependent on the SKU



Q: Is there any multivalued attribute?
A: No

Q: Is there any multiple PKs?
A: No
Q: Is there any partial dependency of an Attribute on the PK?
A: No

Q: Is there any transient dependency of A non-key attribute on another non-key Attribute?
A: No. Result: Table is in 3NF



DATA NORMALIZATION TABLES

⑥ ***Q:*** Is there any multivalued attribute?
A: No

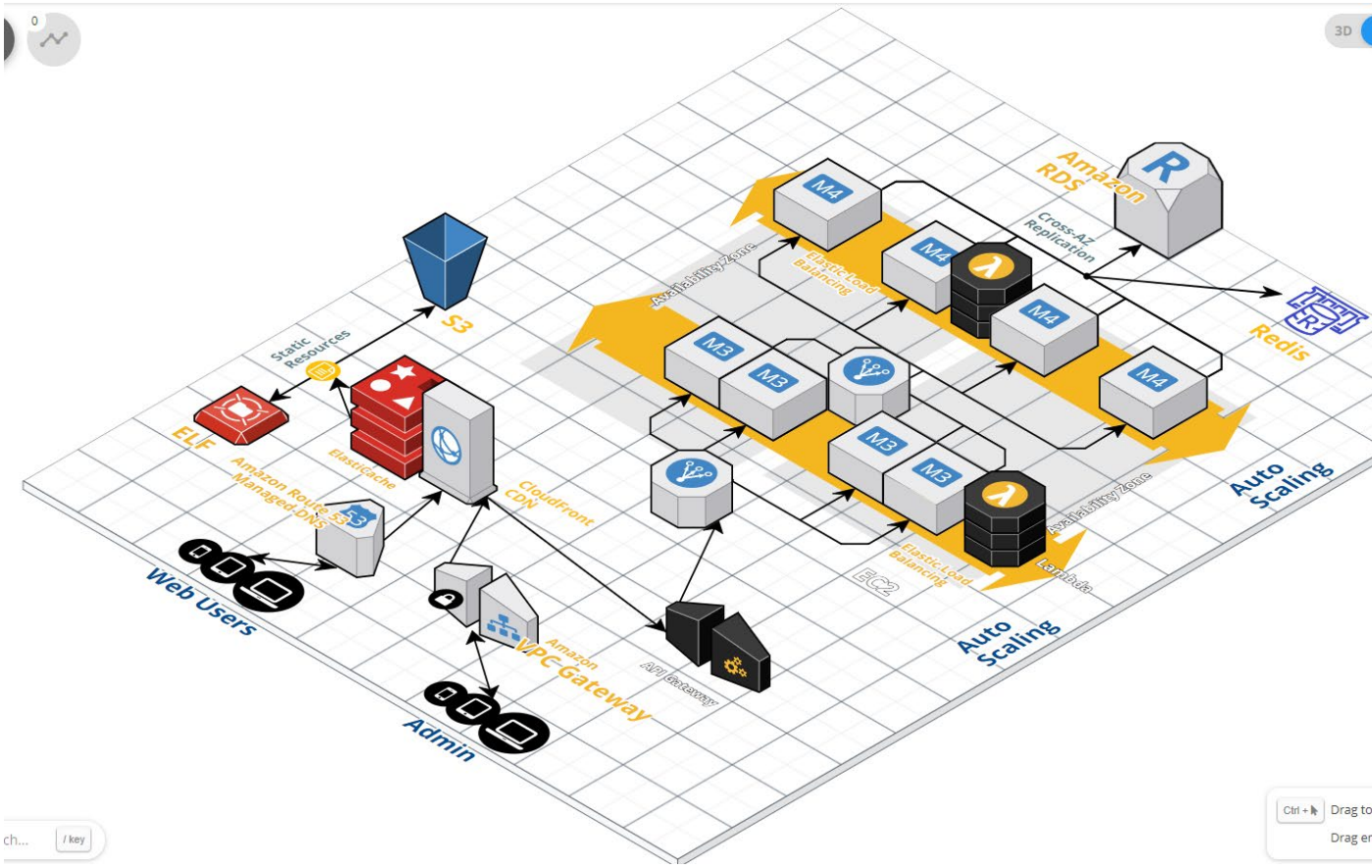
Q: Is there any multiple PKs?
A: No
Q: Is there any partial dependency of an Attribute on the PK?
A: No

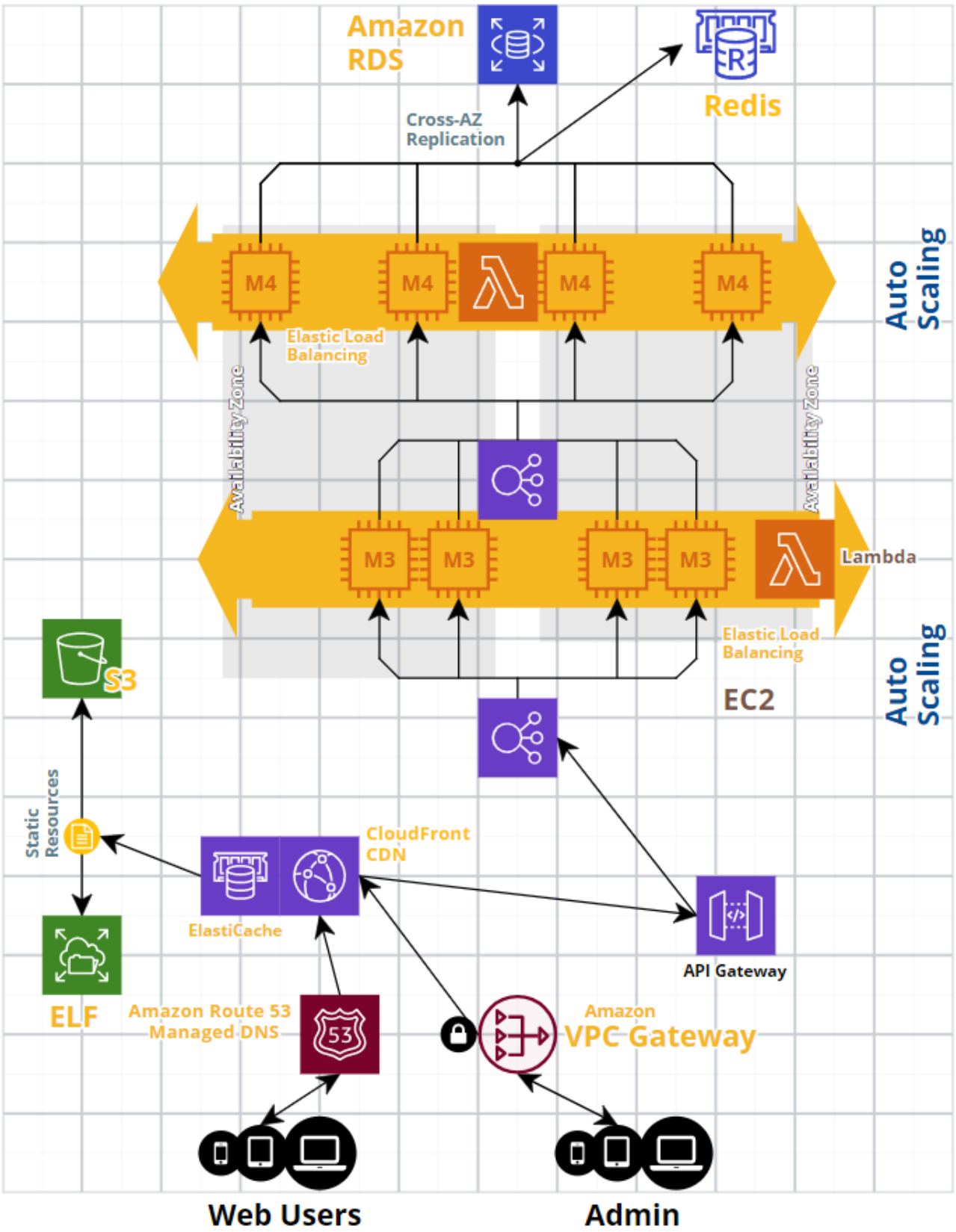
Q: Is there any transient dependency of A non-key attribute on another non-key Attribute?
A: No
Result: Table is in 3NF

Order Fulfilment
OrderFulfimentID(PK)
OrderID:int (FK)
Full Name:string
CustomerBarcode:int
BillingAddress:array
OrderID:int
OrderDate:timeDate
ShippingDate: timeDate
OrderStatus: Order Status
:

TASK 3: ARCHITECTURE DESIGN

3-D Model of the Deployment Diagram for Thompson Cycles





2-D Model of the Deployment Diagram for Thompson Cycles

FEATURES OF THE THOMPSON CYCLES DEPLOYMENT DIAGRAM USING AWS ARCHITECTURE

The Network Diagram was drawn using an industry-standard architecture tool called Cloudcraft specifically tailor-made for Amazon Web Services (AWS) Architectures.

The Thompson Cycles system leverages on the advanced and sophisticated frameworks for Clouding Computing using Amazon Web Services Serverless Capabilities.

As stated in the previous assessment, Amazon Web Services (AWS) offers an extensive, cost-effective Process-as-a-service (PaaS), Infrastructure-as-a-service (IaaS) and Software-as-a-service (SaaS) that offers digitalized scalability and availability (Amazon Web Services 2019).

The Cloudcraft helps you to calculate the cost of your architecture as you build, allowing you to design and scale with both functionalities and cost in view.

In view of the highly modularized, low-coupling deployment-enabled environment, we devised a thin-Client-Server Architecture.

This tremendous technology will no doubt reposition Thompson Cycles to compete intelligently and out-muscle rivals in a most cost-competitive way.

AWS MICROSERVICES EXPLAINED

Amazon Web Services provides a vast array of services that has completely reshaped the paradigm of application development and deployment (Narula et al). The concept of modularizing development and deployment architectures comes with very strong benefits that can be utilized for the Thompson Cycles Project.

FEATURES AND ADVANTAGES

High Modularization and Organizational Flexibility

As opposed to monolithic architectures, the microservices model uncouples the single-service, tightly-knitted processes by assembling them individually into an application using lightweight APIs (Amazon Web Services, 2019). This offers two advantages; the whole architecture doesn't have to be scaled if just a part of the application is experiencing an uptick of demand. Furthermore, deploying trial features are easier and safer since the failure of a single component won't adversely affect the whole system.

The benefits are staggering for an enterprise such as Thompson Cycles and the Managed Extensibility Framework Incremental Model for developing the System: It offers high agility and independence for project teams, application diversity, scalability and maintainability.

Serverless Architecture

Amazon Web Services provides user-adjustable, computing and database building blocks of fully managed Server Infrastructure that eliminates tasks such as hardware purchases provisioning, backup operations, recovery and other maintenance operations related to owning an on-site server (Amazon Web Services, 2019). That means the entire Data Layer and Application Logic Servers, Storage facilitations and Networking functionalities are operated as a service by AWS.

The features selected based on the Non-Functional Requirements for Thompson Cycles are as follows:

DATA STORAGE AND DATA ACCESS LOGIC

Amazon Simple Storage Service (S3) and *Elastic File Storage* and is deployed to store files in its scalable directory that delivers data transfer capability over the internet. To further accelerate the speed and performance of the system to achieve our non-functional requirement, I factored into my design a cashing system comprising of an in-memory data store structure called *Redis AWS ElastiCache*, to cache data with a sub-millisecond data accessibility response time that eradicates seek and conversion time delay from the traditional Disk or Solid State Databases, and *CloudFront Content Delivery Network (CDN)* that compliments data delivery from region to region.

The Redis Database tool also offers asynchronous back-up through a system of master-slave replication that not only greatly improves read operations but also serve as back-up facility in case there is a server outage (Amazon Web Services, 2019). I paired this facility with the Amazon Relational Database storage using a Microsoft SQL Server access application layer to read and write the data.

APPLICATION LOGIC

As clearly stated in the Non-Functional Requirement, we are leveraging on auto-scaled, Elastic Load Balancing (ELB)-empowered *Amazon Elastic Cloud Compute (EC2)* and *Amazon Lambda* which offer the Windows environment for my ASP.NET Framework Languages utilizing the Common Runtime Language to design, run, host and serve the domain/business logic for Thompson Cycles.

PRESENTATION LOGIC

I am deploying a progressive web application that enables the web users to enjoy the full functionalities of a native app through their web browsers eliminating the need to visit an app store. For easy access, a pop feature that invites the user to add web app shortcut to their home screen to their Home Screen is creatively implemented.

The presentation tier uses a web browser to load Graphical User Interface using client-side programming languages such as HTML5, CSS and JavaScript. This layer communicates with the

application tier on the Server-side using the Amazon API Gateway. I plan to embed ASP.NET directives into the HTML script to inject innovative and functional dynamism into the webpage.

NETWORKING AND SECURITY

I have made a strategic selection of Amazon Route 53 and Amazon Virtual Private Cloud (VPC) network structure to deliver a world-class networking model.

Amazon Route 53 furnishes web user with network connection to the Thompson Cycles application facilities domiciled within the Amazon Web Services ecosystem while providing a routing mechanism to other external infrastructures as well. Route 53 Traffic Flow Visual Editor boasts of DNS health management--control, monitoring and configuration-- tools that allows internet traffic to be routed to healthy system endpoints (Amazon Web Services, 2019). This ensures the system is a fault-resistant, low-latency architectural design.

Additionally, a "maintenance and development" network architecture is used to buttress the system using the Amazon Virtual Private Cloud. The VPC provides a private-facing networking workbench where IP address range can be controlled, subnet can be created and, route tables and network gateways can be designed. Some aspects of the backend systems and servers can even be placed in a private-mode, internet-less subnets.

HTTPS SECURITY

I am offering an industry-standard internet communication encryption using HTTP over Transport Layer Security to ensure the privacy, authentication and authenticity of data transfer during client-server exchanges.

AN HIGHLIGHT OF THE NON-FUNCTIONAL REQUIREMENTS (FROM PREVIOUS ASSESSMENT) FOR THE DEPLOYMENT DIAGRAM

1. System must be compatible with Mastercard, Google Pay, Apple Pay, PayPal and Visa Checkout.
2. Websites must be optimized for mobile and tablets.
3. System must support **iZettle Reader** payment plugin and terminal.
4. Compatible with barcode scanner using Bluetooth, Wi-Fi or wired connection.
5. Order Management System and Point-of-Sale will work in Windows Environment.
6. System must use both server cache and browser cache to improve Order Management

7. System speed, website speed and Search Engine Optimization scores.
8. System must be able to support seasonal, flash-sale and peak period heavy traffic using Amazon Web Services' (AWS) Infrastructure-as-a-Service platform to scale servers.
9. System must use Redis other than MySQL to handle Repeated Search Queries.
10. System must use Amazon EC2 Auto-scaling, AWS Elastic Load Balancer and Amazon Elastic File System.
11. A maximum Page Load Speed of 2 seconds and minimizes redirects.
12. System must have Address Verification System (AVS) that verifies the billing address against the issuing banks address.
13. System must Comply with Payments Card Industry Data Security Standards (PCI DSS).
14. System must carry logo of the company.

TASK 4: UI DESIGN PRINCIPLES & USER INTERFACE MOCK-UP

Using the Brand Name: "No Asthma", I crafted the User Interface for the Legal Entity: "Thompson Cycles" by deploying a well-balanced and thoughtfully engineered design philosophy.

LAYOUT

While the designing the No Asthma Interface, I used sophisticated modelling tools with functionalities such as grids, snap-to-Object Geometry, Rulers and Layout Makers to ensure that I achieve a sterling information architecture and strong visual characteristics for the applications.

I utilized a well-aligned, visually-balanced, consistently white-spaced layout to create an info-graphically stimulating page. Overall, my contents were arranged, positioned and interlinked for easy accessibility and smooth navigation.

FUNCTIONAL AESTHETICS

Interface should inspire interaction (Stylianidis 2015). A visually appealing functionality was drawn to inspire navigation, evoke product appreciation, and command visual appeal that immerses the user into No Asthma Interface with the goal of improving their experience and usability. Animations of action button, drop-down selection and navigation icons are created not only to be eye-catching but also catalyse site interface awareness.

CONTENT AWARENESS

The No Asthma Interface is a well networked, interconnected, inter-navigable site. I ensured that I utilized page header to inform the user what the page is about, complemented by a visual hierarchy styling that helps concentrate the user on the most important aspect of the page.

Different visual treatment was also given to certain areas to help the user graphically appreciate the different but related aspects of a page information and feature.

Visual Cues are embossed into the site to show page bearing, the user browser trails and page interconnection links that helps user navigate to other adjacent and related pages. This is to prevent the isolation of a page from other aspects of the site or application that the user might be immediately interested in.

In line with the most recent content awareness techniques (Williams 2019), navigation and functionality pop-ups are used to bring to the awareness of the user certain functionality in order to improve the effectiveness of the site.

CONSISTENCY

The idea of "consistency" is differentiative uniformity and thematic similarity (Porter n.d.). User interface elements that are behaviourally and functionally diverse must never be undifferentiated in the name of consistency as this will adversely impact on usability (Babich 2019). Conversely, oneness

of layout groupings, similarity of text area, and themes of colour palette and font design are utilized for similar pages, headers, features and contents of the No Asthma Interfaces.

MINIMAL USER EFFORT

The minimalization of user efforts was cleverly and ingeniously embedded into every single interface of the No Asthma Web Applications. For example, as demonstrated in the UI Prototypes, a user can add or proceed to basket from the home page or from any other pages on the web application with just one click; making it intuitively easy to navigate to checkout. Adding items to the basket couldn't be any easier with just one click.

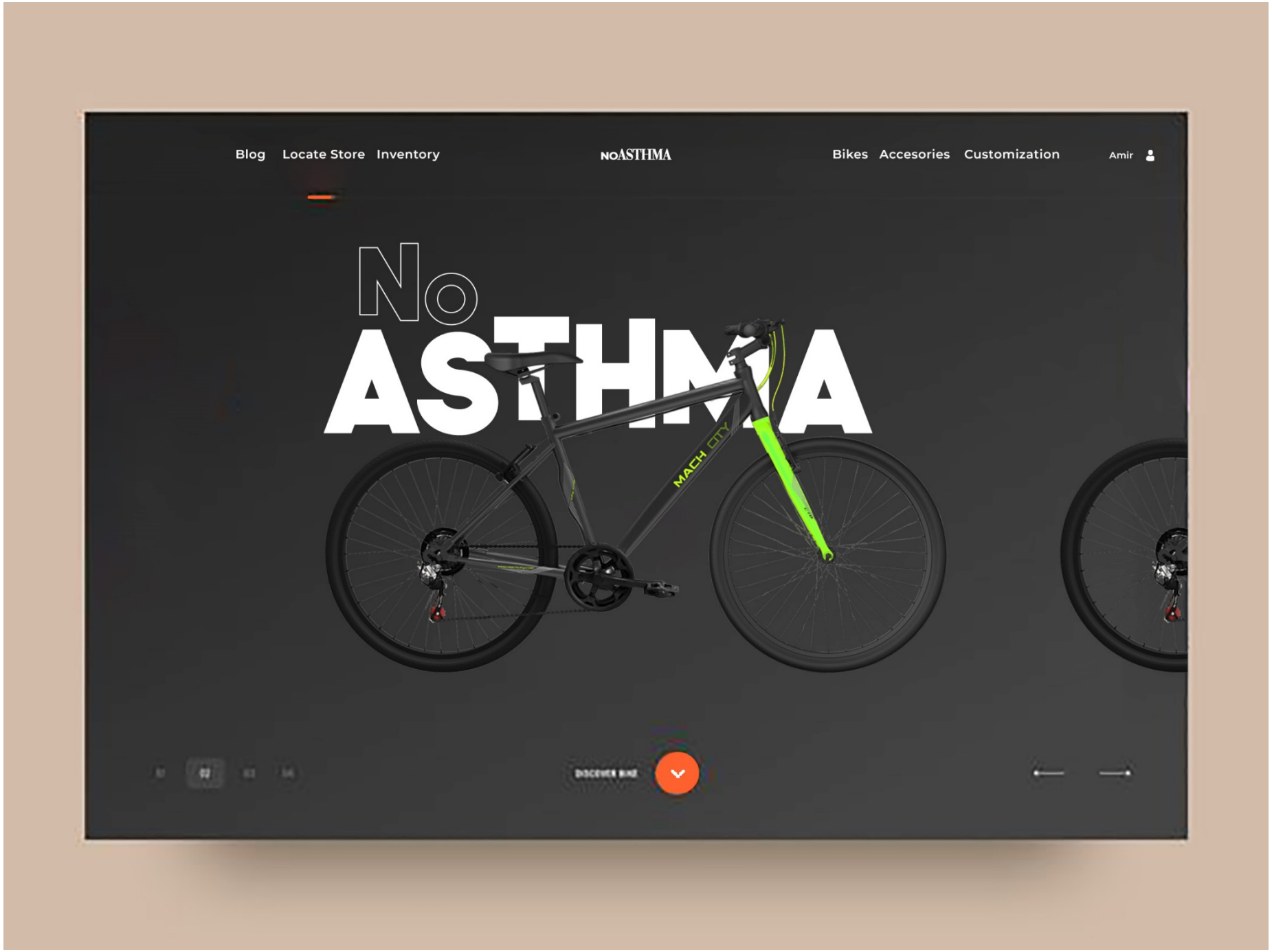
On the checkout page, it takes only one click to confirm payment, provided the user account or devices ware initially set up with bank cards or a pre-installed Google Pay, Apple Pay or PayPal respectively.

USABILITY

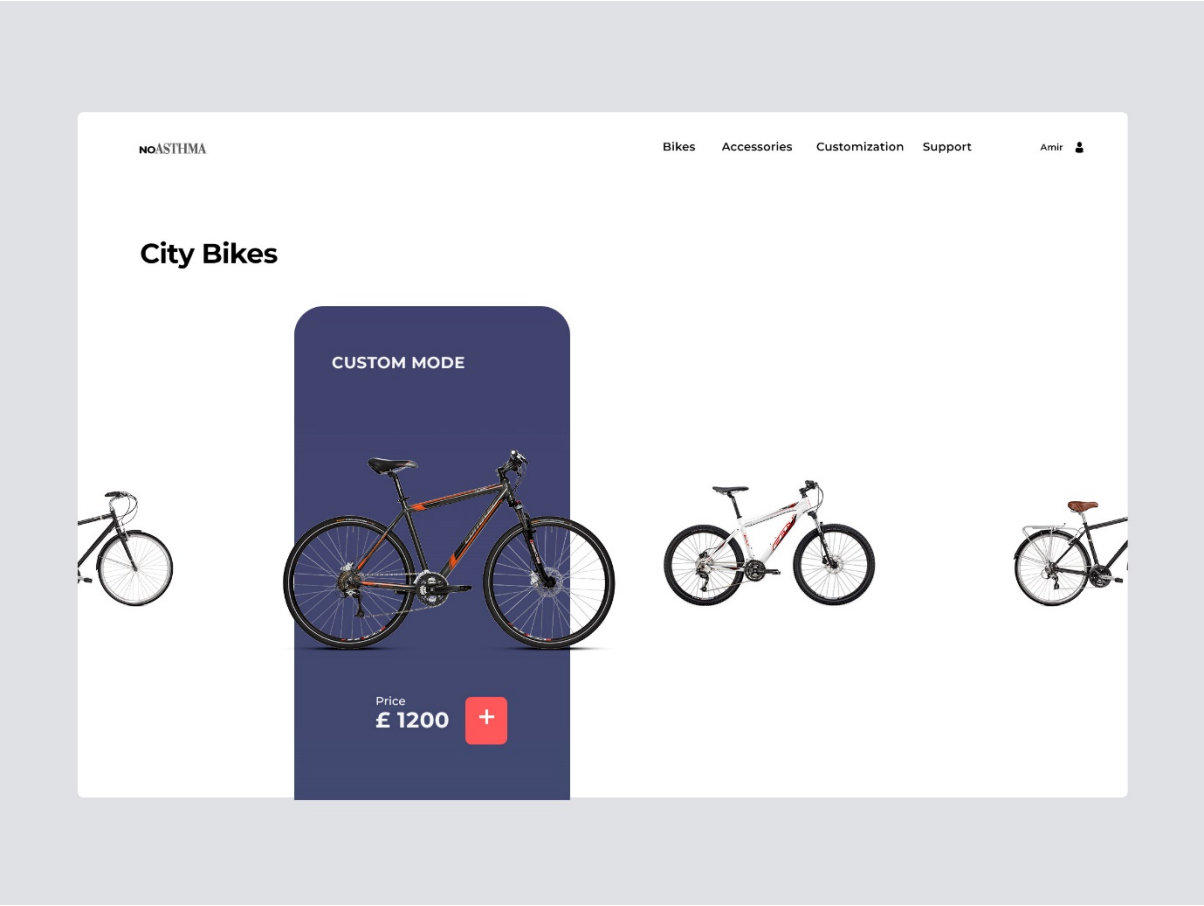
The Design of the No Asthma is a usability-driven design: the application was not purposed for superfluous minimalism or unnecessary complexity but instead a thoughtful, careful consideration for the user objectives, needs and goals was ensured. Observable from the UI Prototypes, here is what I did: I struck a fine balance between ease of use and ease of learning. This I have done by providing top-level app functionalities without compromising simplicity or advancing the complexity of usage.

I will discuss more on this point at the Usability and User Experience Goals in Task 6.

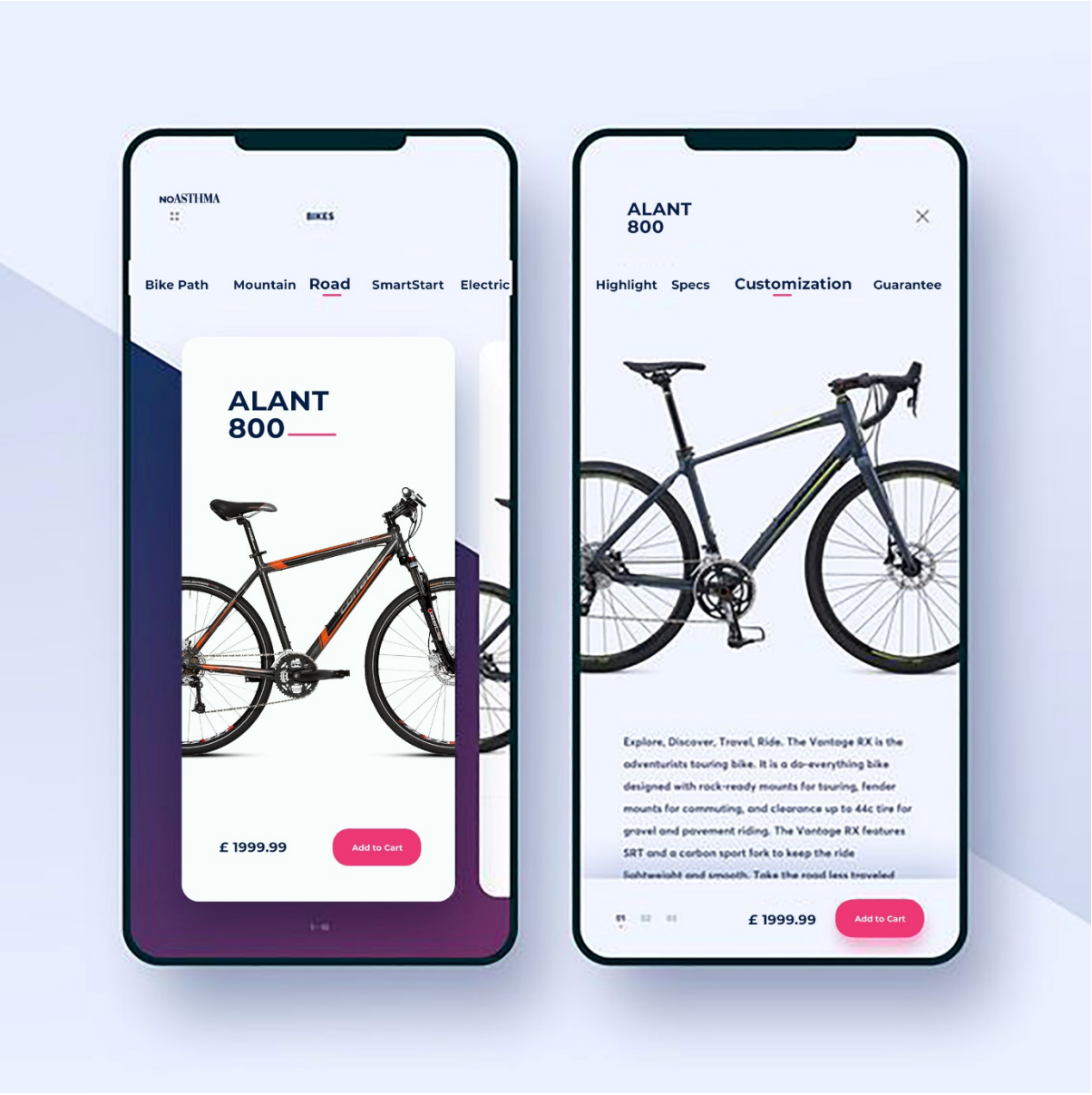
NO ASTHMA HOME PAGE: DESKTOP VIEW



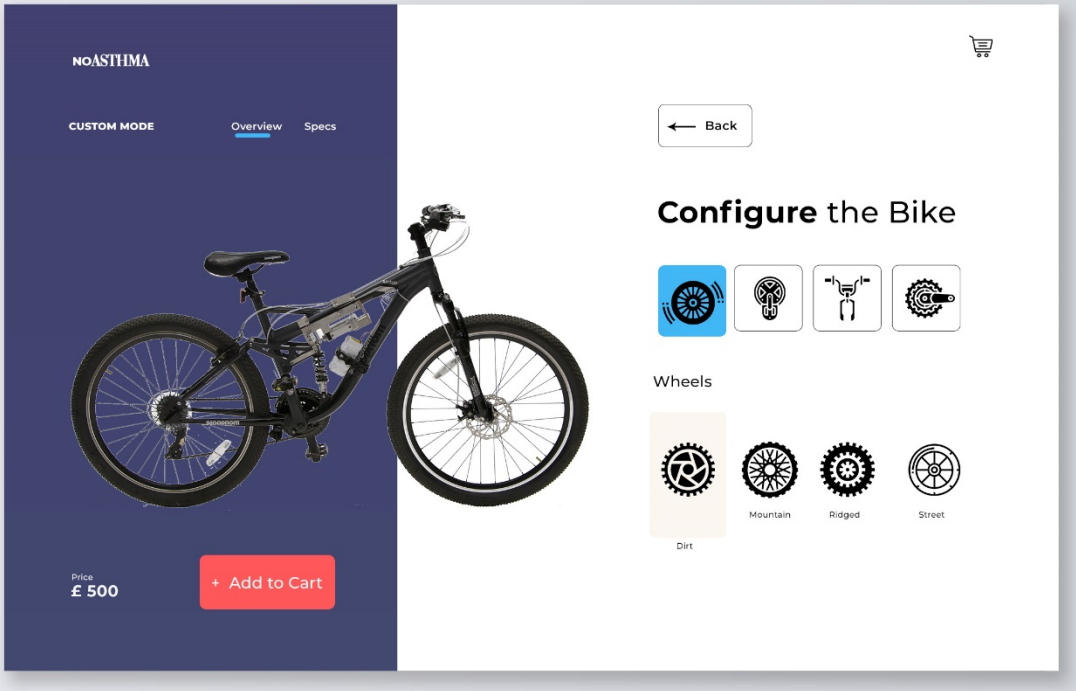
NO ASTHMA BIKES DISPLAY: DESKTOP VIEW



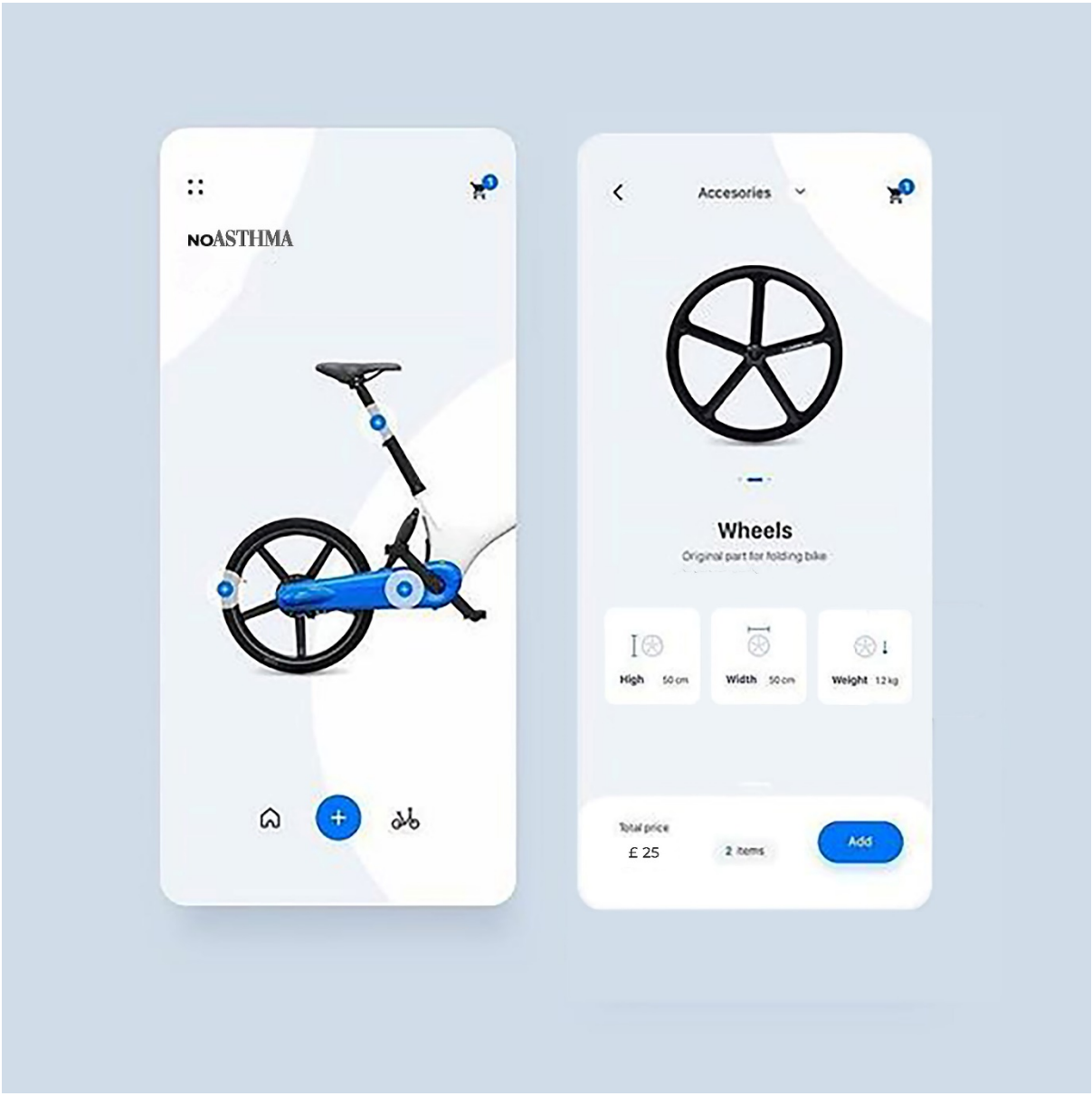
NO ASTHMA BIKES DISPLAY: MOBILE VIEW



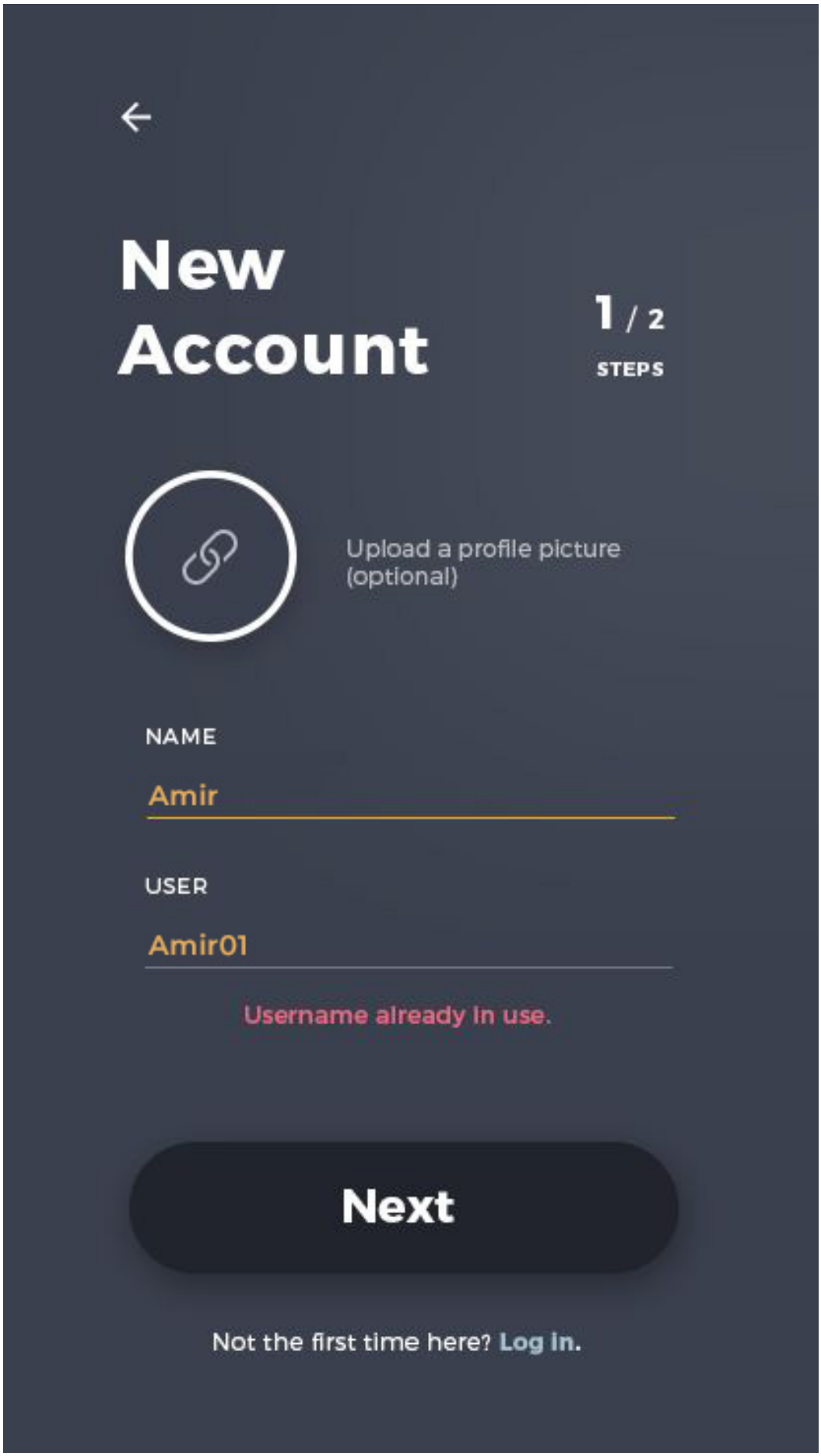
NO ASTHMA BIKE CONFIGURATION PAGE: DESKTOP VIEW



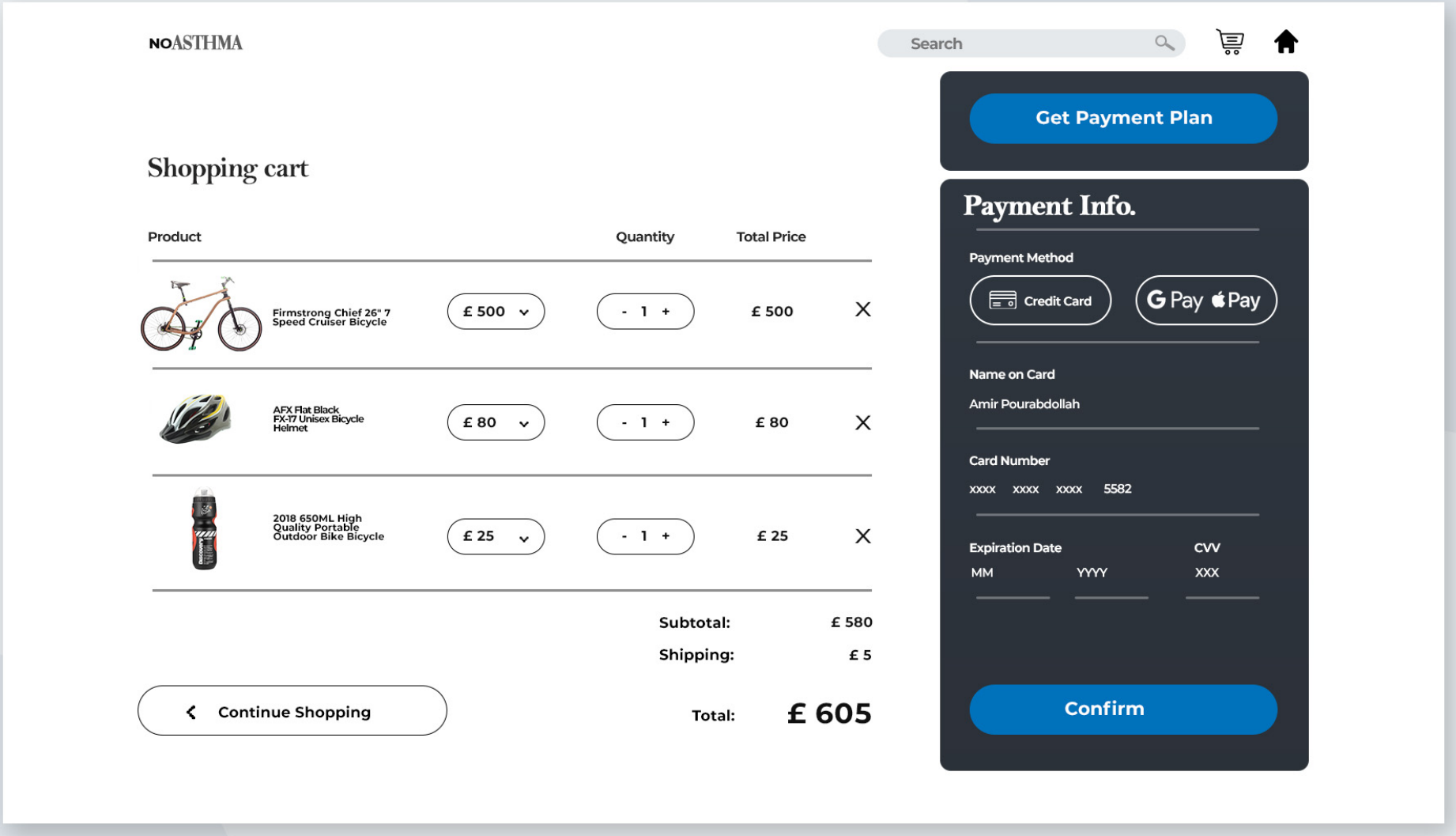
NO ASTHMA BIKE CONFIGURATION PAGE: MOBILE VIEW



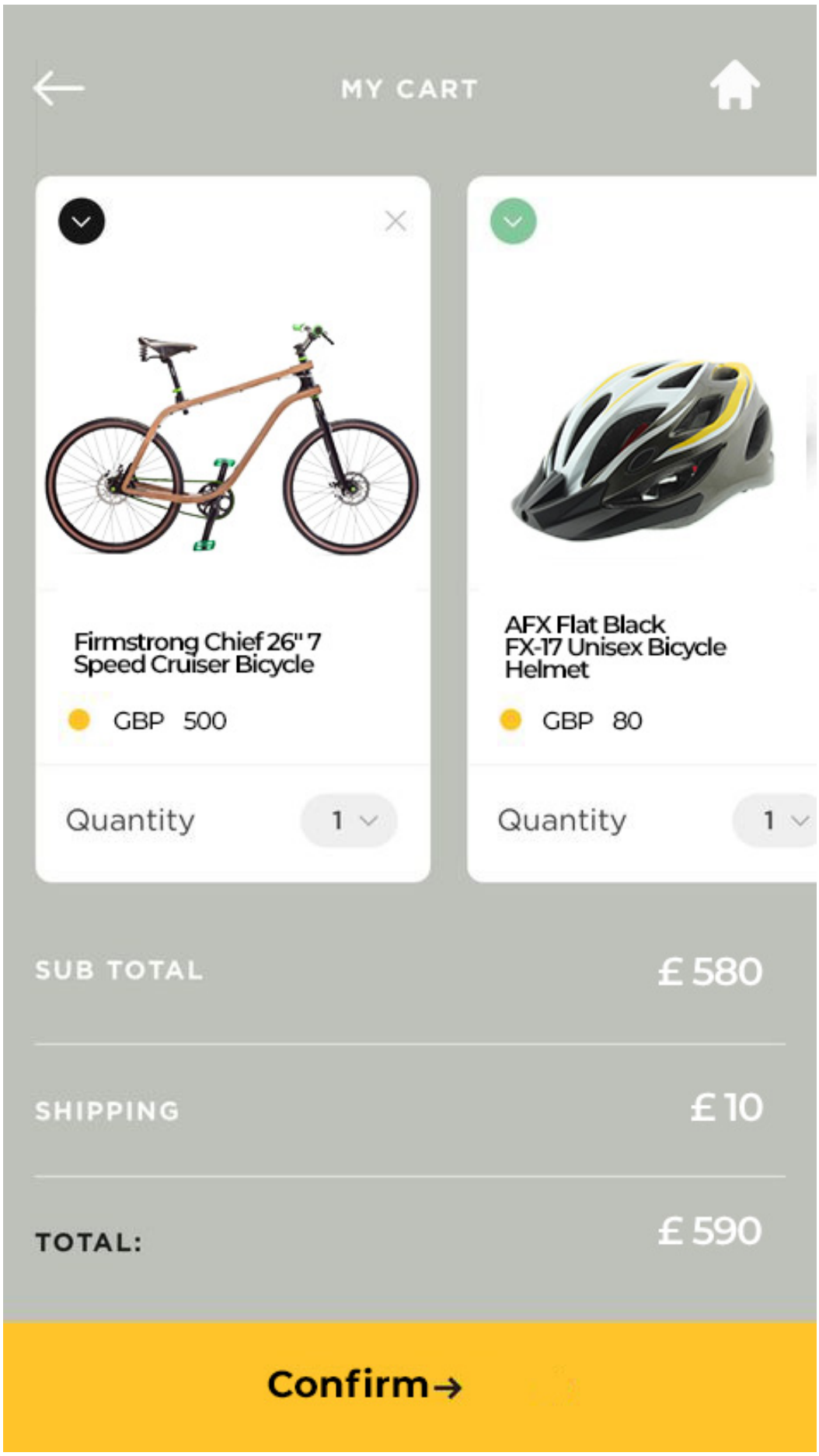
NO ASTHMA SIGN UP: MOBILEVIEW



NO ASTHMA CHECKOUT PAGE : DESKTOP VIEW



NO ASTHMA CHECKOUT PAGE: MOBILE VIEW



TASK 5: UI DESIGN PROCESS

USE SCENARIOS

Unregistered User creates an Account

1. The user loads the No Asthma web home page.
2. The user clicks on the sign-up button.
3. The user fills the required fields such as name, user name card details and shipping details.
4. The user confirms the e-mail address.
5. The user receives a successful completion notification.

Registered User Successfully Completes an Order

1. User loads the No Asthma Web home page.
2. User adds items to cart.
3. User views order summary.
4. User gets payment plan.
5. User proceeds to make the first instalment.
6. User receives notification for successful payment.

UI STANDARDS UTILIZED IN THE NO ASTHMA USER INTERFACE DESIGN

In a visual system, UI Metaphors and Templates play a vital role of linking concepts in human language and the business domain to the application domain in a way that greatly increase user familiarity and understandability of the System (Alexandros 2015). As supported by the work of (Abdulhassan et al 2011), the following Visual Metaphors were utilized for the No Asthma website:

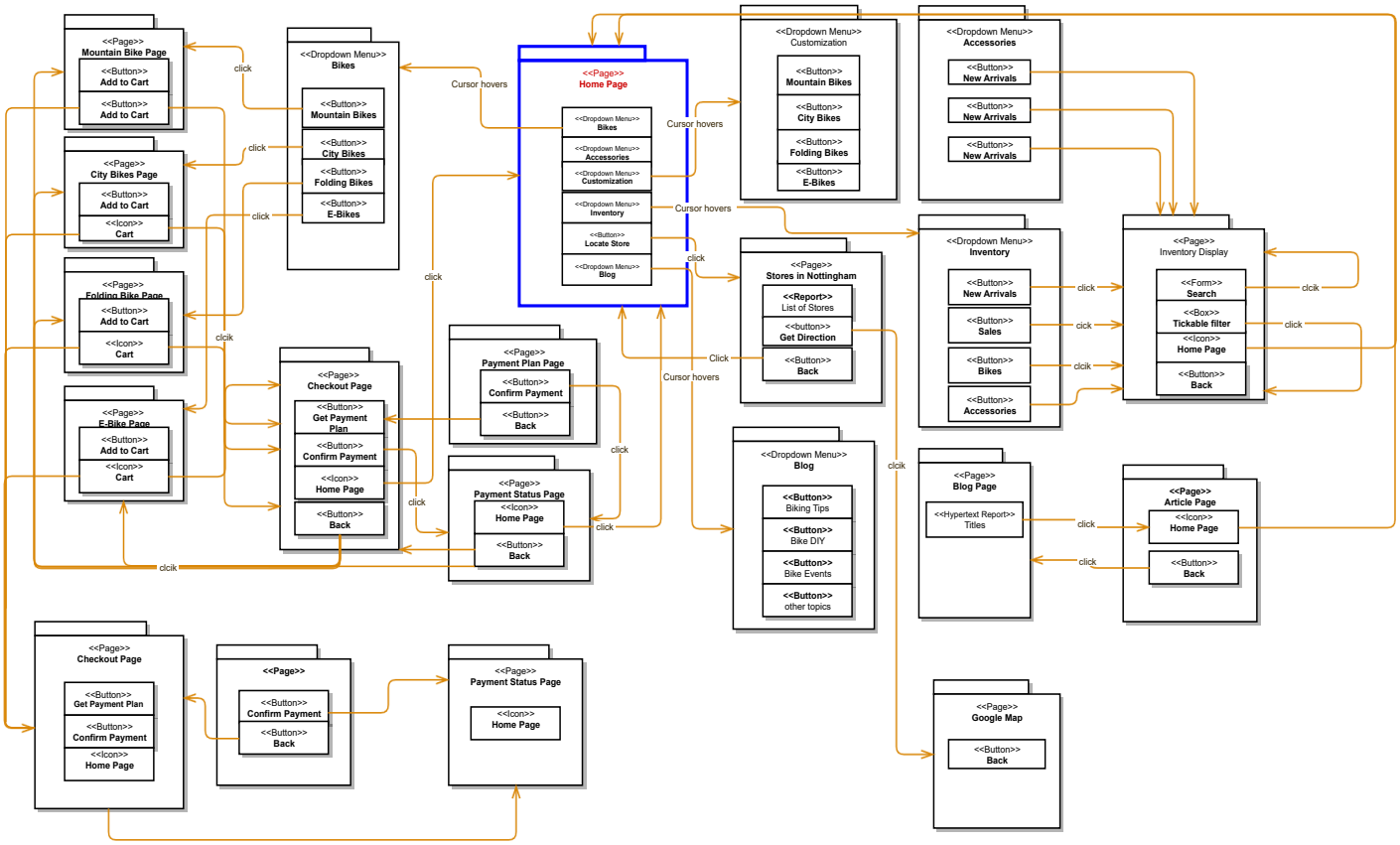
1. *Home Icon:* This typifies the front or approach view of a House. This was metaphorically deployed in pages that are away from the home page. This can greatly improve the navigation experience.
2. *Slide Icon:* This suggests to the user that there are other pictures to be viewed in the slides
3. *Shopping Cart:* Drawing on the visual representation of the cart in a shopping mall, this inspires the users of the web app, to add items to the cart, view items and order summary from the cart. This is very analogous of the operations in the user is most likely used to in a physical setting.
4. *Back and Forward Arrows:* These arrows are used with the slide icon to help the user navigate the direction of views. Also, the web browsers come with the arrows to help assist the user in the navigation of the pages; indicating previous back pages and previous front pages respectively.
5. *Letter "X":* This is a graphical representation of cancellation. This is cleverly deployed to indicate to the user that they can delete an item from the cart.
6. *Conversation Pop Up:* As discussed earlier, informational pop ups that points to and briefly explains functionalities that can easily go unnoticed.
7. *Buttons:* Buttons bring to live to the user the real-world idea of activation of a process. Buttons were used to figuratively depict to the user that he is about to trigger an event. This is used for the sign-in/sign-up action, the items of the dropdown menu of the headers, checkout, get payment plan and confirm payment actions.
8. *Highlighter and Animations:* The Highlights colour and the Animations *enable the texts, pictures, buttons or icons* under the cursor on the No Asthma Web App to give the impression and strong suggestion it is actionable by clicking. This makes the system more intuitive and responsive to the user engagements. These were used for the displays that can be expanded for more info, and for texts that are hyperlinked to other pages.
9. *Hourglass:* This is suggesting to the user using the time concept of an hourglass that a process is underway.

NO ASTHMA WEB APP PROTOTYPE AND EVALUATION TECHNIQUE

Using the Jakob Nielsen's (1994) Heuristics for the appraisal of the No Asthma Web App User Interface, the following perspectives were considered:

1. *Transparency of System Status:* The No Asthma Web App should always keep user's abreast of the system process activated by the user. For example, an animated hourglass was used as a metaphor to signal to the user that card transaction is being processed, with an immediate feedback on the result of the process. This is a rule of thumb.
2. *Use of Metaphors:* This is a way of making the Thompson Cycles systems speak the user's language with graphical representations, phrases, words and expressions that are familiar to the user.
3. *Consistency of UI Standards:* We followed system behaviours users expect. No gimmicky redirects were used, buttons, Icons and Highlights the same thing through the system.
4. *Error Prevention:* I used this for the form submission, custom election, order summary, sign-up and sign-in sessions. The Sign-up button or the sign-in buttons aren't activated until all the required fields are entered, e.g. passwords and username for the sign-in sessions.
5. *Recognition:* Navigation elements are made visible using the animations and colour highlights. It is very intuitive and easy to go back and forth on the No Asthma Web App.
6. *Error Management:* Anticipated errors were communicated to the user in plain language instead of using complex terminology. If a page that doesn't exist is loaded by the user, a simple message is issued by the system to the user, suggesting a few simple actions.
7. *Aesthetics and Minimalist Design:* The information was not cluttered and jam-packed on pages with overstatements and unnecessary expressions. I used secondary, transient pop ups to indicate to the user a new or unnoticed feature and a hover-before-display technique to show additional product info only when the cursor hovers over an icon and product.
8. *User Control and Freedom:* A very great deal of freedom and ownership was conferred to the user. Firstly, being a web app, the user can customize the No Asthma the look and feel of the web app. Secondly, user-specific features such as search history, order history, delete, undo and redo are appended to the user accounts.
9. *Help and Documentation:* Pop-up Prompts were utilized to help users get context-related or task-specific helps. Frequently asked questions were also designed as a quick link for the home page No Asthma Web Page
10. *Efficiency of Use:* I designed a very learnable yet advanced in functionality. Functionalities that might be unfamiliar to novice users are intuitively guided with info-pops and animations that make engaging the content for any class of user interesting.

NAVIGATION DIAGRAM FOR THE NO ASTHMA WEB APPLICATION


<<Page>>
Mountain Bike Page
<<Button>>
Add to Cart
<<Button>>
Add to Cart
<<Page>>
City Bikes Page
<<Button>>
Add to Cart
<<Icon>>
Cart
<<Page>>
Folding Bike Page
<<Button>>
Add to Cart
<<Icon>>
Cart
<<Page>>
E-Bike Page
<<Button>>
Add to Cart
<<Icon>>
Cart
<<Dropdown Menu>>
Bikes
<<Button>>
Mountain Bikes
<<Button>>
City Bikes
<<Button>>
Folding Bikes
<<Button>>
E-Bikes
<<Page>>
Home Page
<<Dropdown Menu>>
Bikes
<<Dropdown Menu>>
Accessories
<<Dropdown Menu>>
Customization
<<Dropdown Menu>>
Inventory
<<Button>>
Locate Store
<<Dropdown Menu>>
Blog
<<Dropdown Menu>>
Customization
<<Button>>
Mountain Bikes
<<Button>>
City Bikes
<<Button>>
Folding Bikes
<<Button>>
E-Bikes
<<Dropdown Menu>>
Accessories
<<Button>>
New Arrivals
<<Button>>
New Arrivals
<<Button>>
New Arrivals
<<Page>>
Stores in Nottingham
<<Report>>
List of Stores
<<button>>
Get Direction
<<Button>>
Back
<<Dropdown Menu>>
Inventory
<<Button>>
New Arrivals
<<Button>>
Sales
<<Button>>
Bikes
<<Button>>
Accessories
<<Page>>
Inventory Display
<<Form>>
Search
<<Box>>
Tickable filter
<<Icon>>
Home Page
<<Button>>
Back
<<Page>>
Checkout Page
<<Button>>
Get Payment Plan
<<Button>>
Confirm Payment
<<Icon>>
Home Page
<<Button>>
Back
<<Page>>
Payment Plan Page
<<Button>>
Confirm Payment
<<Button>>
Back
<<Page>>
Payment Status Page
<<Icon>>
Home Page
<<Button>>
Back
<<Dropdown Menu>>
Blog
<<Button>>
Biking Tips
<<Button>>
Bike DIY
<<Button>>
Bike Events
<<Button>>
other topics
<<Page>>
Blog Page
<<Hypertext Report>>
Titles
<<Page>>
Article Page
<<Icon>>
Home Page
<<Button>>
Back
<<Page>>
Checkout Page
<<Button>>
Get Payment Plan
<<Button>>
Confirm Payment
<<Icon>>
Home Page
<<Page>>
<<Button>>
Confirm Payment
<<Button>>
Back
<<Page>>
Payment Status Page
<<Icon>>
Home Page
<<Page>>
Google Map
<<Button>>
Back
click
Cursor hovers
Click
clcik
cick

TASK 6: USABILITY AND USER EXPERIENCE GOALS

For the Thompson Cycles Applications, using the Software Quality Requirements and Evaluation (SQuaRE) by (ISO 2014), the following project-relevant requirements were factored as key design considerations:

USABILITY GOALS

MAINTAINABILITY

Scalability: From the choice of development methodology to proposed deployment architecture, scalability of software product and support infrastructure (storage, servers and networks) was a major usability factor. From an enterprise perspective, the scalability and changeability of software components and support system is a strategically and economically important to the Thompson Cycles Firm.

Testability: As you would notice, there is a high usage of modularization both in development and deployment in the whole scheme of design in this project, owing to the need to satisfy and stimulate testability factors such as isolate-ability, understandability, controllability, automatability and controllability, aided by the choice of user-controlled, digitally-observable Elastic Cloud Compute(EC2) facilities provisioned by Amazon.

FUNCTIONALITY

Suitability: From the No Asthma Customers, to both Admin and the Business Owner, this system not only offers a comprehensive and significant business process automation but also improvements of business activities such as sales, advertisements, payments, task management and general administration.

Interoperability: The choice of web application for this project was made to greatly aid the interoperability and accessibility across all device and platform types without having to download any additional native apps from the app store--a simple widget saved on the screen will do.

Security: Attention to detail was not spared when it came to the security features of the proposed system. The following components were assigned to the applications: 1. Private Virtual Cloud (VPC) 2. Hypertext Transfer Protocol Secure Over Transport Layer Security 3. Compliance with Payments Card Industry Data Security Standards (PCI DSS). See Task 3 above for details.

EFFICIENCY

Resource-efficiency: From an enterprise perspective, the cost of product development will be strategically limited with the use of a project-specific Hybrid Incremental Model (Using Managed Extensive Framework). Also, using the Amazon Cloud Computing facilities, economies of scale is achieved by Thompson Cycles a share of usage of ultra-large and powerful computing facilities. This is proven to cut cost significantly for development, deployment, operations and maintenance of the applications and infrastructures.

Time-efficiency: Infrastructurally, ample investment has been proposed to improve the performance of the Web Apps. Research has proven that Users abandons apps or sites that they consider slow. The use of In-memory ElastiCache for caching and Lambda Function to serve user requests is geared towards improving the user experience when interacting with the applications.

Additionally, in order to minimize user effort, the three-click rule was extensively employed.

USABILITY

Fault-Tolerance: The web application will reflect a sensitivity towards the tendency of customer user mistakes when reversing a highly important tasks such as deletion of a cart, cancellation of an order or, in case of the Admin, deletion of Reorder, deletion of any report and deletion of inventory.

Intelligibility: The behaviour of the system is styled to user expectations and geared towards meeting user operational needs. Intrusive advertisement pop-ups are not permitted. Sudden, unexpected interruptions are disallowed. Additionally, the design principles put into use will ensure that the system can be easily learned and understood by the users.

USER EXPERIENCE GOALS

DESIRABLE GOALS

1. Intuitive
2. Useful
3. Engaging
4. Immersive
5. Supportive and Transparent
6. Productive
7. Personalization-controls

UNDESIRABLE GOALS

1. Intrusive
2. Annoying
3. Deceptive
4. Disorganized
5. Difficult to understand
6. Difficult to navigate
7. Dysfunctional

REFERENCES

1. Amazon Web Services, 2019. *Amazon Web Services – Overview of Amazon Web Services.* Seattle: Amazon. Available at: https://d1.awsstatic.com/whitepapers/aws-overview.pdf [Accessed 8 December 2019].
2. Stylianidis, 2015. Metaphors in User-Interfaces Design. Birmingham: University of Birmingham Available at: https://pdfs.semanticscholar.org/fe80/67cde778efd8e8406e3a3fa24b0afb766979.pdf [Accessed 8 December 2019].
3. R. S. Abdulhassan, M. Masood and A. Hosam, 2011. "Semiotic Differences of Macintosh Os X & Microsoft Windows 7 Based on Metaphors and Interpretation," *2011 Second International Conference on Intelligent Systems, Modelling and Simulation*, Kuala Lumpur, 2011, pp. 209-213. DOI: 10.1109/ISMS.2011.41 [Accessed 8 December 2019].
4. ISO, 2014. *Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — Guide to SQuaRE*[online]. Geneva: ISO. Available at: https://www.iso.org/obp/ui/#iso:std:iso-iec:25000:ed-2:v1:en [Accessed 8 December 2019].
5. J. Porter, n.d. http://bokardo.com/principles-of-user-interface-design/ [online]. Available at: http://bokardo.com/principles-of-user-interface-design/ [Accessed 10 December 2019].
6. N. Babich, 2019. *The Four Golden Rules of Design* [online]. San Jose: Adobe Incorporations. Available at: https://xd.adobe.com/ideas/process/ui-design/4-golden-rules-ui-design/ [Accessed 8 December 2019].

7. J. M. Williams 2019. *Ten Essential UI Design Tips* [online]. San Francisco: Webflow. Available at: https://webflow.com/blog/10-essential-ui-design-tips [Accessed 10 December 2019].

8. S. Narula, A. Jain and Prachi, 2015 "Cloud Computing Security: Amazon Web Service," *Fifth International Conference on Advanced Computing & Communication Technologies*, Haryana, 2015, pp. 501-505. DOI: 10.1109/ACCT.2015.20

9. Akamai Technologies, 2019. *Performance Matters: Key Consumer Insights.* Massachusetts: Akamai Technologies. Available at: https://www.akamai.com/us/en/multimedia/documents/content/akamai-performance-matters-key-consumer-insights-ebook.pdf [Accessed 8 December].