Data Diagrams

Data Types

= primary key
= foreign key

userInformation

Field	Data Type
userID	int(11)
bankID	int(11)
accessibilityID	int(11)
first_name	varchar
last_name	varchar
email	varchar
password	varchar
phone_number	varchar
dob	date
prev_ticketIDs	varchar
points	int(11)
staff	varchar

bankingInformation

Field	Data Type
bankID	int(11)
card_number	varchar
card_cvc	varchar
exp_date	varchar
card_holder	varchar
address	varchar

zoo_tickets

Field	Data Type
ticketID	int(11)
userID	int(11)
date_of_purchase	datetime
start_time	datetime
end_time	datetime
ticket_price	float(2, 2)
ticket_type	varchar

zoo_availability

Field	Data Type
date	date
start_time	time
end_time	time
spaces	smallint(3)
ticket_type	varchar

= primary key
= foreign key

Data Types 2

blog

Field	Data Type
blogID	int(11)
creation_date	datetime
title	varchar
caption	varchar
image	varchar

accessibility

Field	Data Type
accessibilityID	int(11)
dark_mode	bool
screen_reader	bool
mono_chromatic	bool

hotel_rooms

Field	Data Type
room_number	int(11)
userID	int(11)
booked	bool
start_time	datetime
end_time	datetime
room_price	float(3, 2)
capacity	smallint(1)
room_type	varchar

hotel_availability

Field	Data Type
accessible	bool
room_number	int(3)
start_date	datetime
end_date	datetime
room_type	varchar

Data Dictionary - userInformation

Field	Data Type	Data Format	Field Size	Description	Example
userID	Integer	x….	11	The user ID	2
bankID	Integer	x….	11	The key to the user's bank information in the bankingInformation table	3
accessibilityID	Integer	x….	11	The key to the user's accessibility information in the accessibility table	4
first_name	Variable Character	-	50	The user's first name	John
last_name	Variable Character	-	50	The user's last name	Smith
email	Variable Character	xxxxx@xxxx.xxx	255	The user's email address	john.smith@gmail.com
password	Variable Character	-	255	The user's password for his/her account	password1234321
phone_number	Variable Character	02x AAAA AAAA	15	The user's phone number	+44 7123 456789
dob	Date	YYYY-MM-DD	-	The user's date of birth	2000-10-31
prev_ticketIDs	Variable Character	x….	11	This is the IDs of the tickets the user has previously purchased	1, 4, 6, 7, 9
points	Integer	-	11	This is the amount of points that the user has accumulated	1234
staff	Variable Character	-	255	This says if they are staff or not, and what type of staff	admin

Data Dictionary - bankingInformation

Field	Data Type	Data Format	Field Size	Description	Example
bankID	Integer	x….	11	The user's bank ID	2
card_number	Variable Character	-	30	The user's card number	4111111111111111
card_cvc	Variable Character	-	5	The user's card cvc	123
exp_date	Variable Character	-	50	The user's card expiry date	10/26
card_holder	Variable Character	-	255	The name on the user's card	Mr. John Smith
address	Variable Character	-	255	The user's billing address	Ian McSheerman, 20 Prince Street, London, SW4 0SB

Data Dictionary - zoo_tickets

Field	Data Type	Data Format	Field Size	Description	Example
ticketID	Integer	x….	11	The ticket ID	2
userID	Integer	x….	11	The user's ID of who the ticket belongs to	3
date_of_purchase	Datetime	YYYY-MM-DD HH:MM:SS	-	When the ticket was purchased	2020-12-31 23:59:59
start_time	Datetime	YYYY-MM-DD HH:MM:SS	-	The start time of the user's ticket	2020-12-31 23:59:59
end_time	Datetime	YYYY-MM-DD HH:MM:SS	-	The end time of the user's ticket	2020-12-31 23:59:59
ticket_price	Float	-	4	The price of the user's ticket	20.34
ticket_type	Variable Character	-	255	The type of ticket it is	Adult

Data Dictionary - zoo_availability

Field	Data Type	Data Format	Field Size	Description	Example
date	Date	YYYY-MM-DD	11	The date of the availability	2020-12-31
start_time	Time	HH:MM:SS	11	The start time of the availability	23:59:59
end_time	Time	HH:MM:SS	11	The end time of the availability	23:59:59
spaces	Integer	-	50	The amount of spaces the availability has	200
ticket_type	Variable Character	-	50	The type of ticket that is available at that time and date	Adult

Data Dictionary - blog

Field	Data Type	Data Format	Field Size	Description	Example
blogID	Integer	x….	11	The blog's ID	2
creation_date	Datetime	YYYY-MM-DD HH:MM:SS	-	The creation date of the blog	2020-12-31 23:59:59
title	Variable Character	-	255	The title of the blog	Red panda
caption	Variable Character	-	255	The text within the blog	This red panda is cool
image	Variable Character	-	255	The hyperlink for the image in the blog	https://upload.wikimedia.org/wikipedia/commons/thumb/e/e6/Red_Panda_%2824986761703%29.jpg/800px-Red_Panda_%2824986761703%29.jpg

Data Dictionary - accessibility

Field	Data Type	Data Format	Field Size	Description	Example
accessibilityID	Integer	x….	11	The ID that links to the user table	2
dark_mode	Boolean	-	-	Tells if dark mode is enabled or not	True
screen_reader	Boolean	-	-	Tells if screen reader is enabled or not	False
mono_chromatic	Boolean	-	-	Tells if mono chromatic mode is enabled or not	False

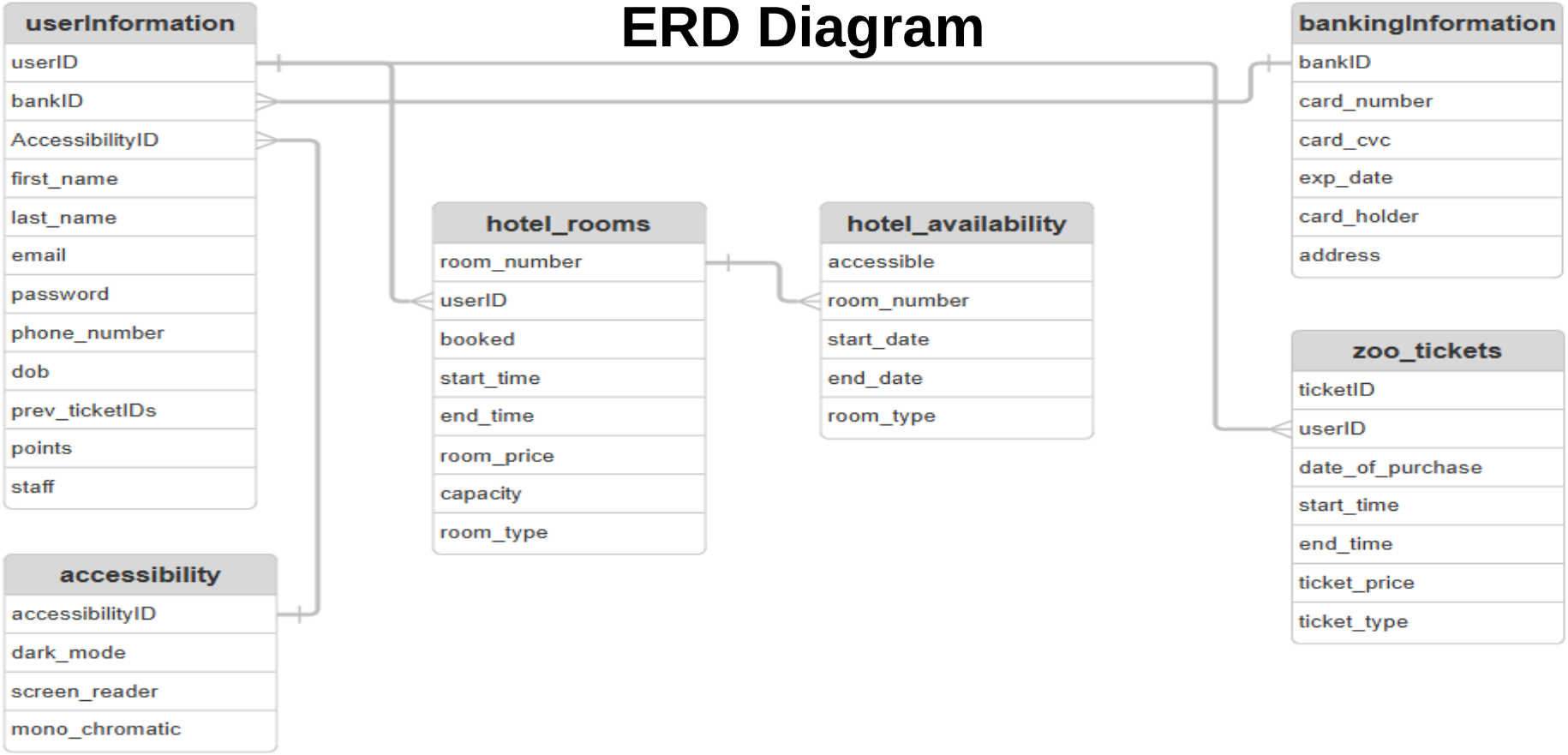
Data Dictionary - hotel_rooms

Field	Data Type	Data Format	Field Size	Description	Example
room_number	Integer	x….	3	The room number	2
userID	Integer	x….	11	The user's ID to who has taken it	3
booked	Boolean	-	-	If the room is booked or not	True
start_time	Datetime	YYYY-MM-DD HH:MM:SS	-	The start staying time for the user who has taken it	2020-12-31 23:59:59
end_time	Datetime	YYYY-MM-DD HH:MM:SS	-	The end staying time for the user who has taken it	2020-12-31 23:59:59
room_price	Float	-	5	The price of the room	123.45
capacity	Small Integer	-	1	The capacity of the room, i.e. how many people it can hold	3
room_type	Variable Character	-	255	The type of room it is	Double Room

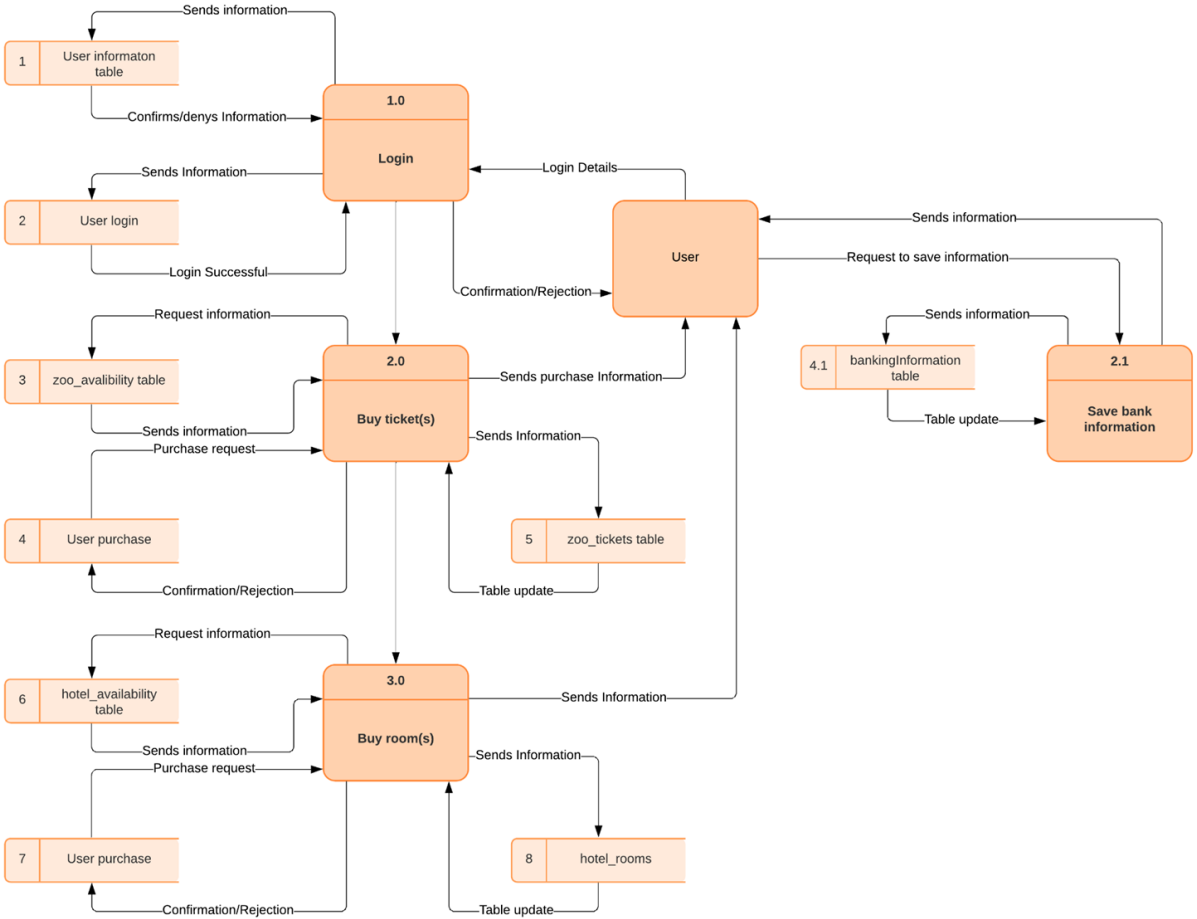
Data Dictionary - hotel_availability

Field	Data Type	Data Format	Field Size	Description	Example
accessible	Bool	-	-	If the room is accessible or not	True
room_number	Integer	x….	3	The number of the room	231
start_date	Datetime	YYYY-MM-DD HH:MM:SS	-	The start date and time when it is available from	2020-12-31 23:59:59
end_date	Datetime	YYYY-MM-DD HH:MM:SS	-	The end date and time when it is no longer available from	2020-12-31 23:59:59
room_type	Variable Character	-	255	The type of room it is	Double Room

ERD Diagram


userInformation
userID
bankID
AccessibilityID
first_name
last_name
email
password
phone_number
dob
prev_ticketIDs
points
staff
accessibility
accessibilityID
dark_mode
screen_reader
mono_chromatic
hotel_rooms
room_number
userID
booked
start_time
end_time
room_price
capacity
room_type
hotel_availability
accessible
room_number
start_date
end_date
room_type
bankingInformation
bankID
card_number
card_cvc
exp_date
card_holder
address
zoo_tickets
ticketID
userID
date_of_purchase
start_time
end_time
ticket_price
ticket_type

Data Flow Diagram



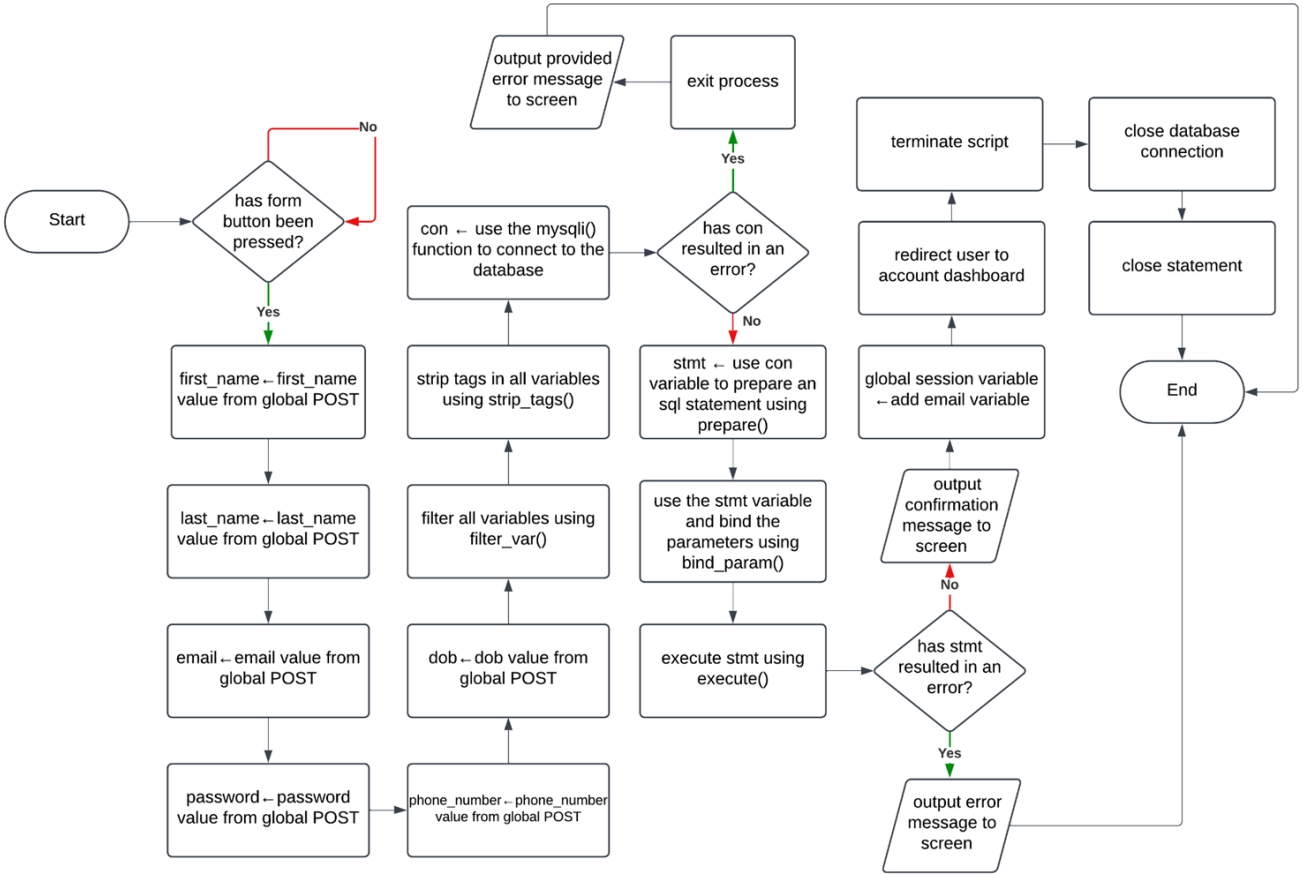
Your algorithm designs do not need to show the whole solution but should show how you would solve a number of different key problems.

You should limit your algorithm designs to a maximum of five complex problems.

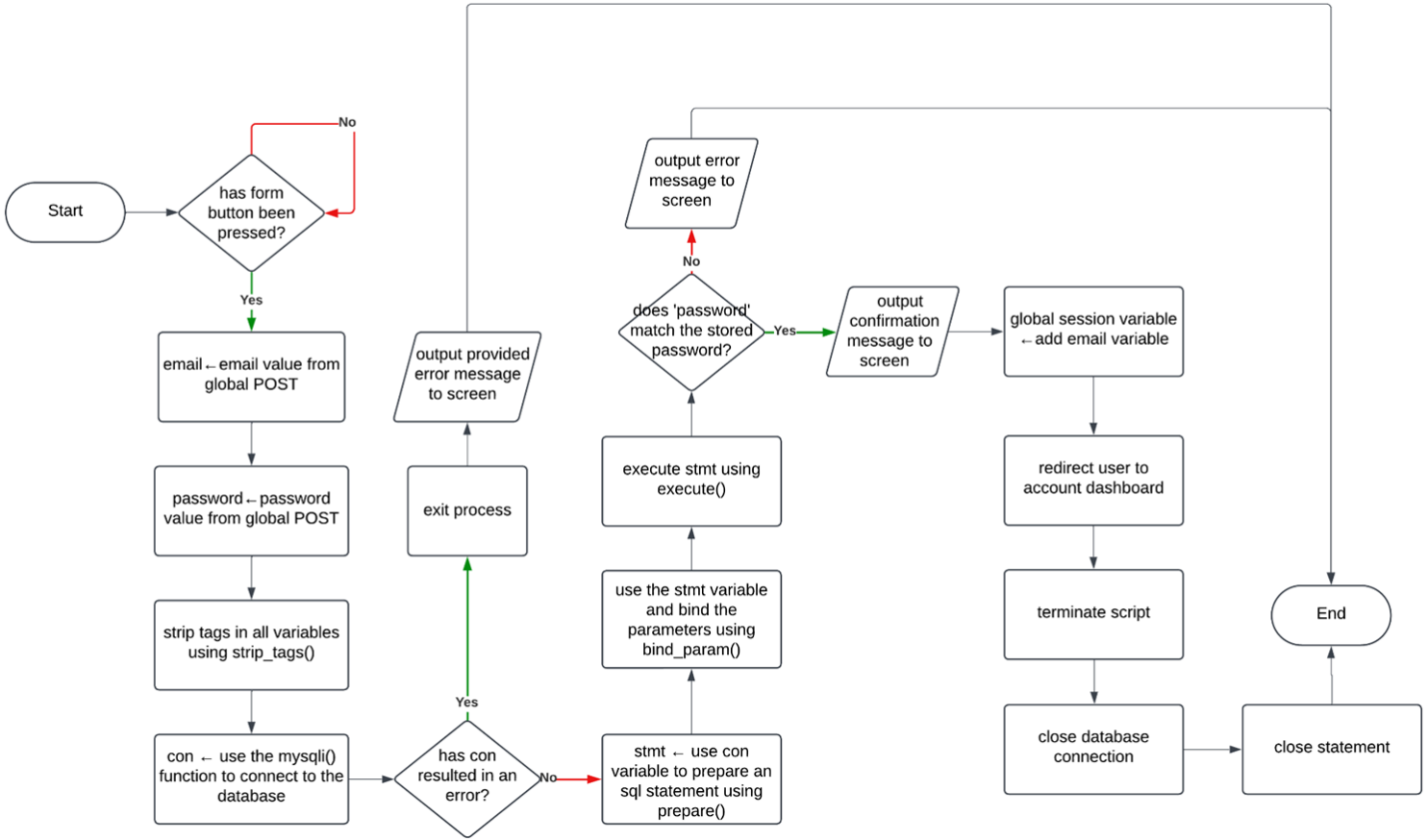
Algorithms

Pseudocode - Sign Up

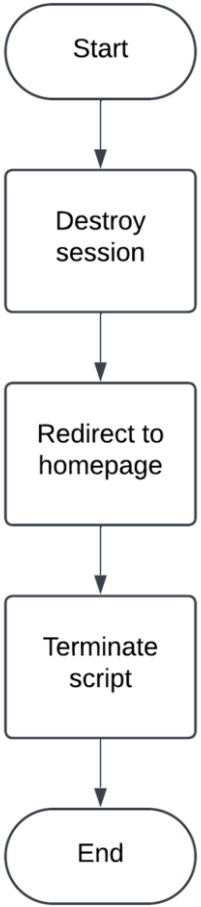
Flowcharts - Sign Up



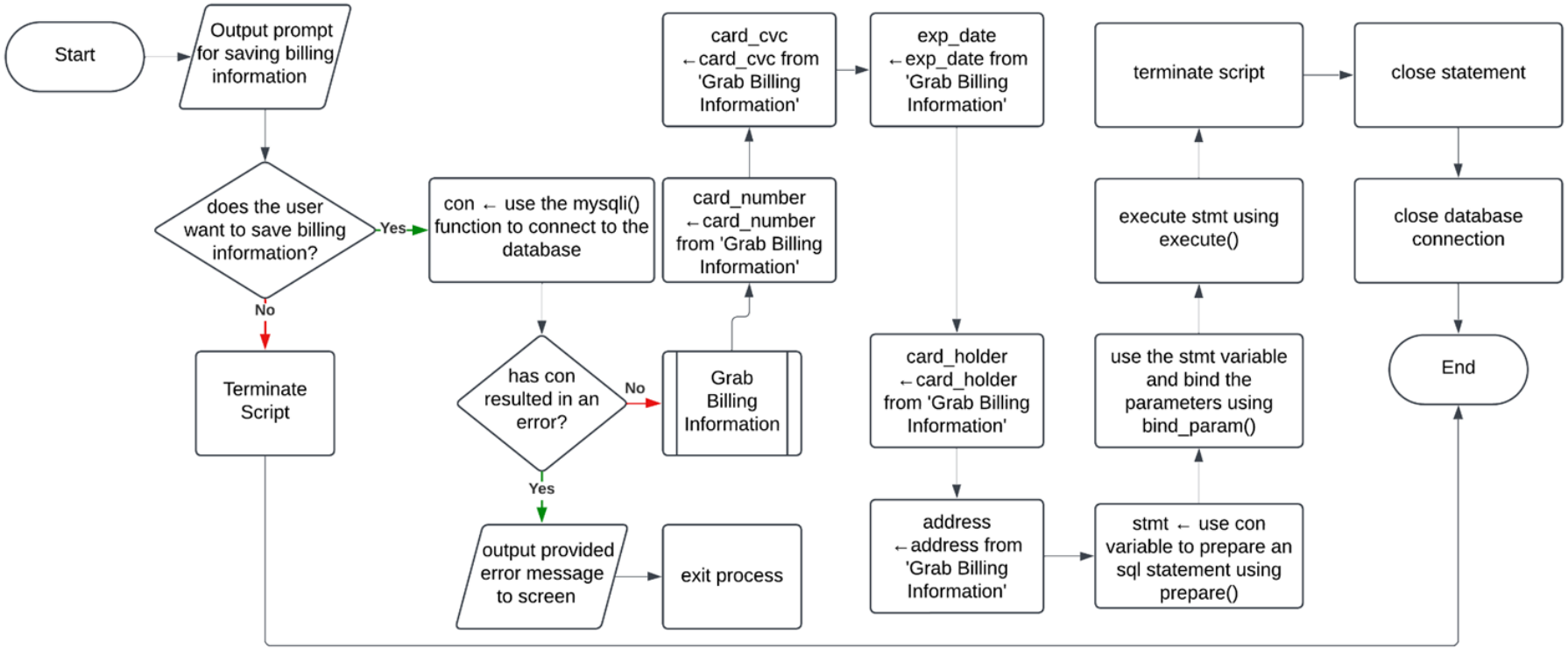
Flowcharts - Login



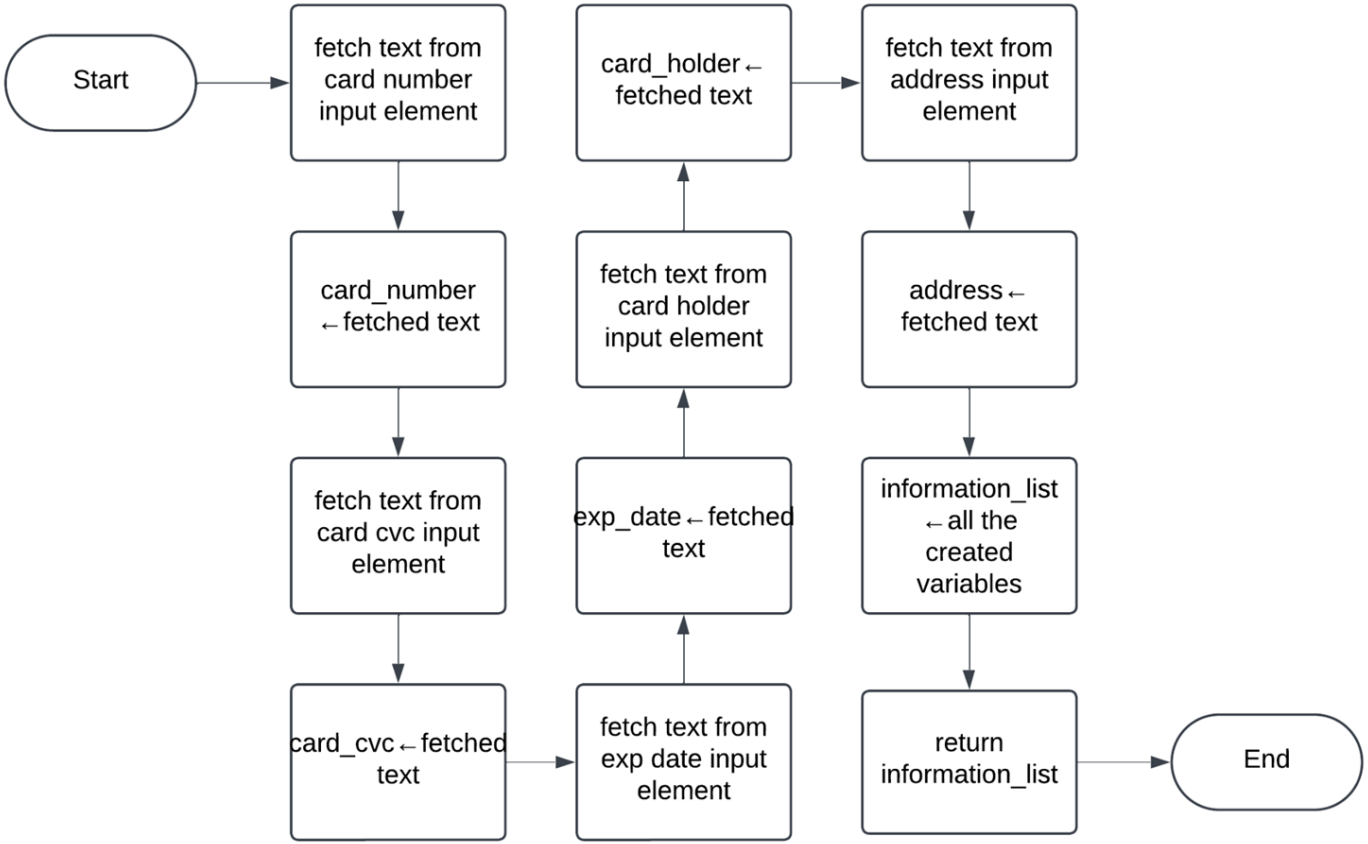
Flowcharts - Logout



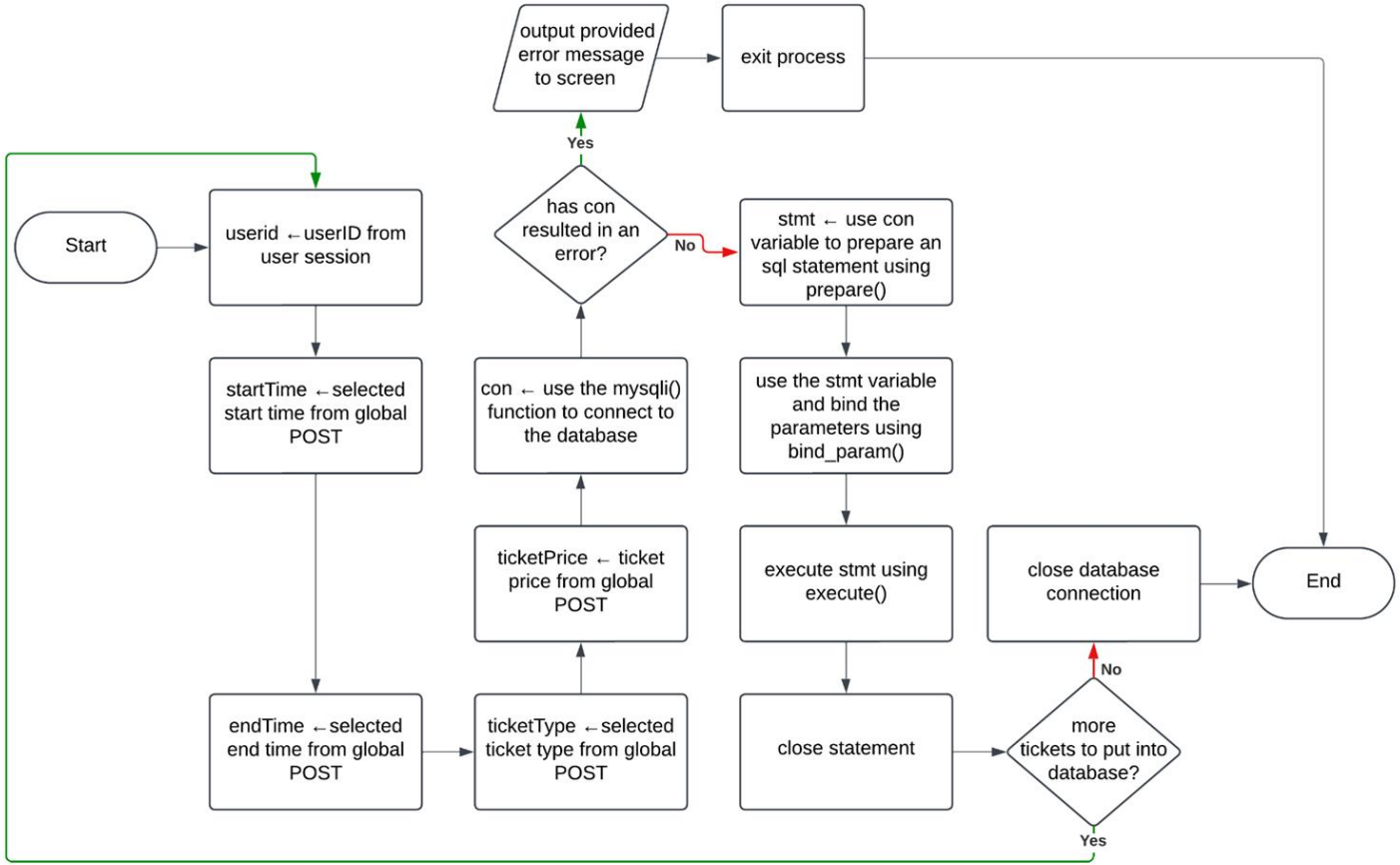
Flowcharts - Saving Billing Information (from checkout)



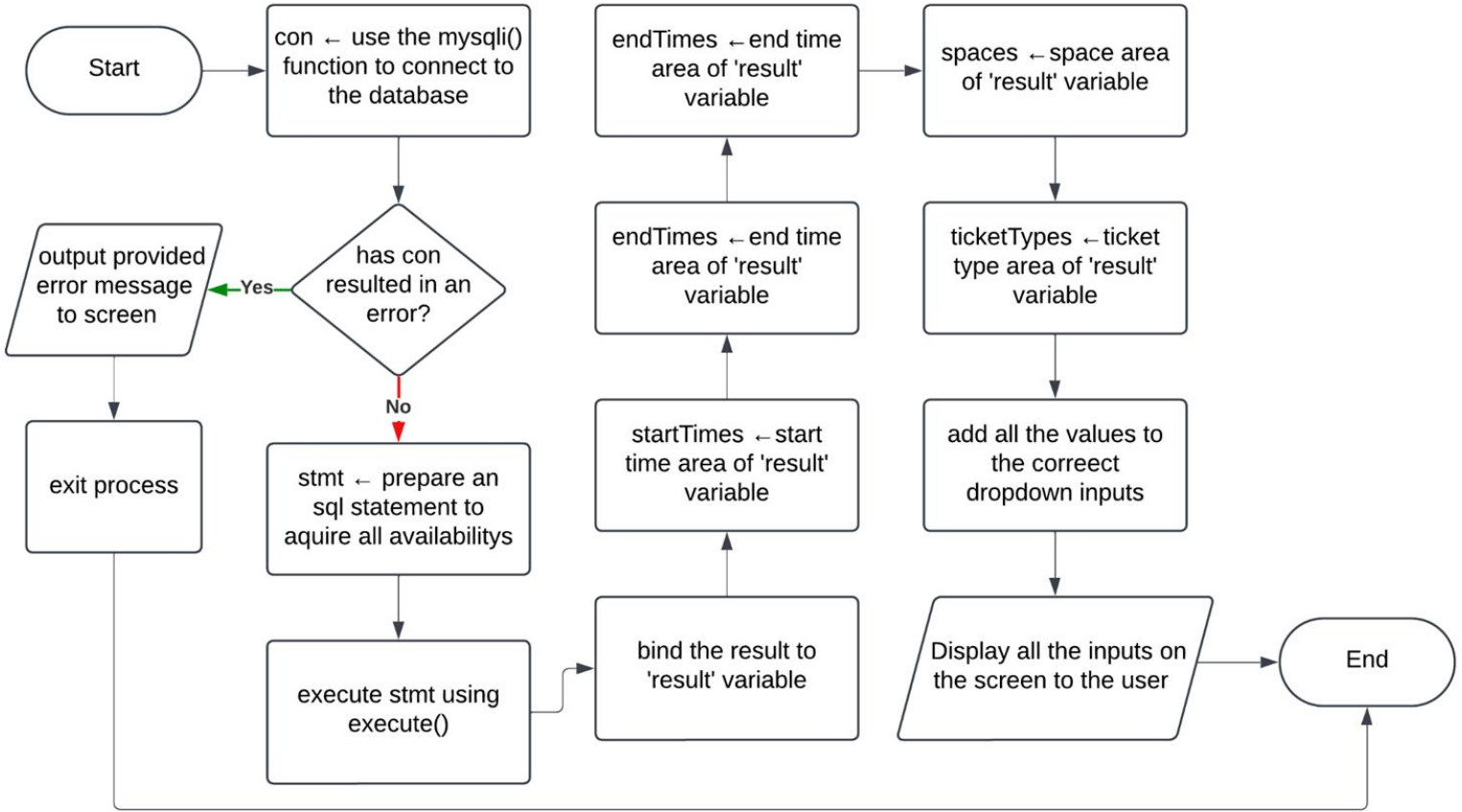
Flowcharts - Grab Billing Information (from checkout)



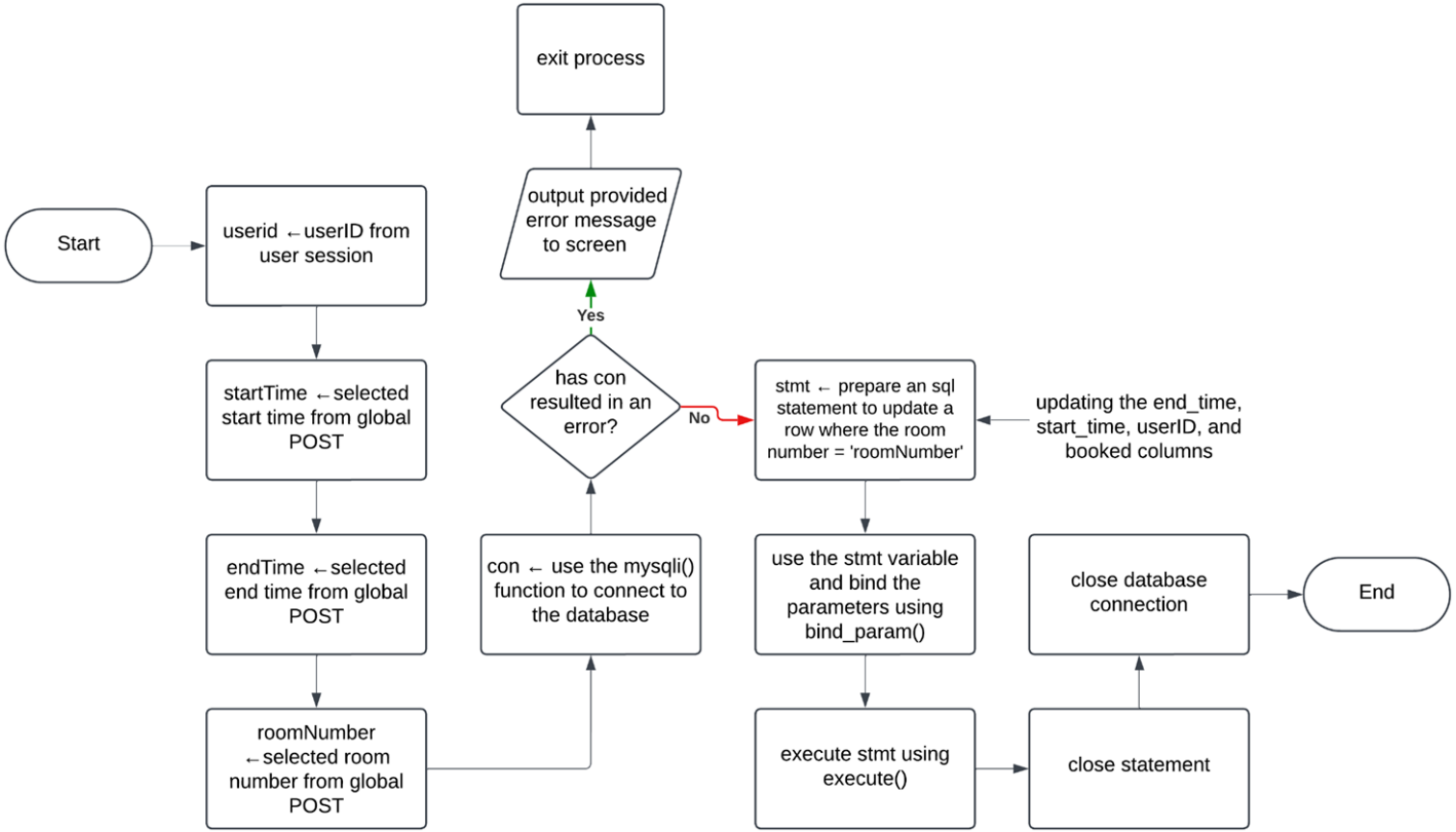
Flowcharts - Booking Zoo Tickets



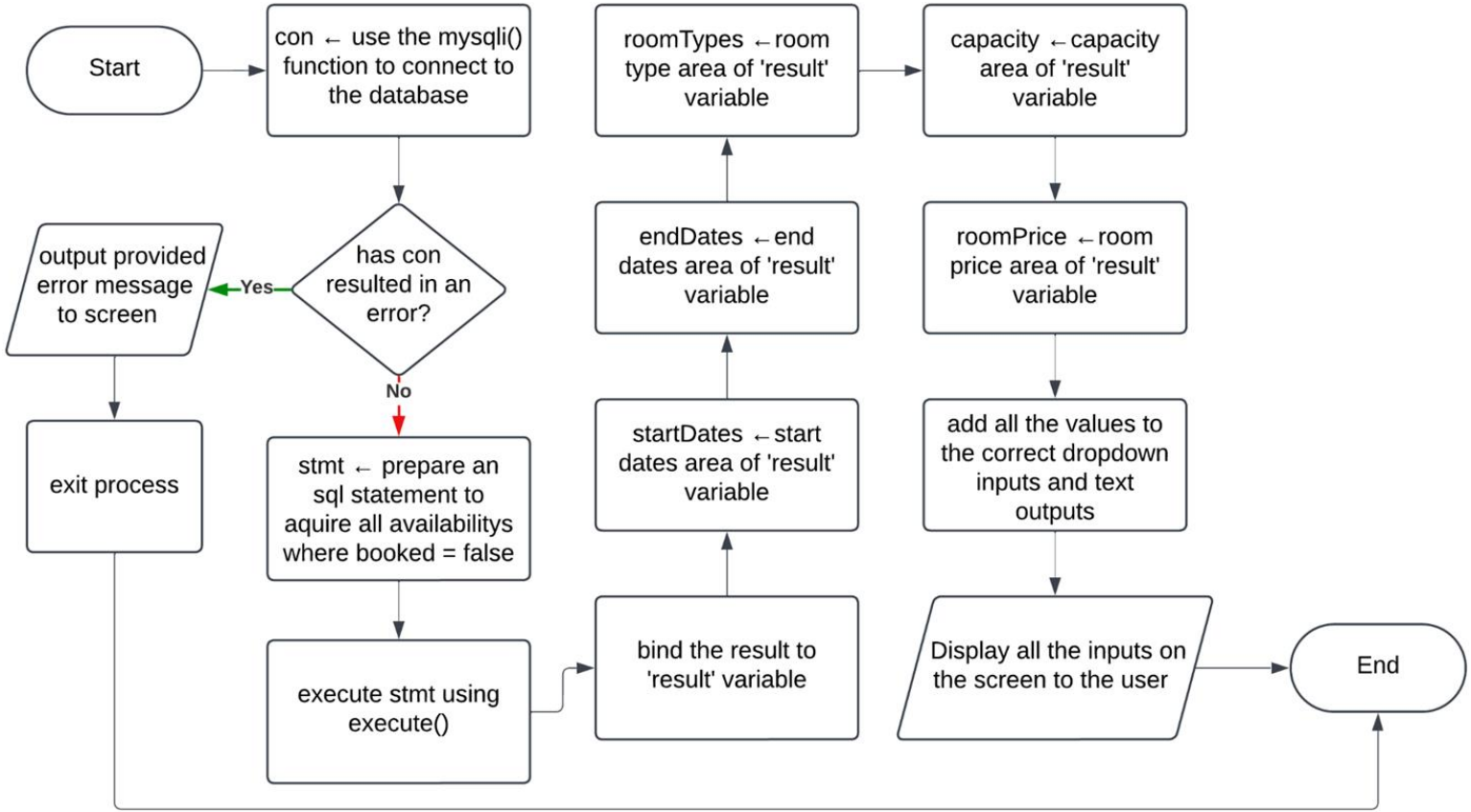
Flowcharts - Display Zoo Availability



Flowcharts - Booking Hotel Rooms



Flowcharts - Display Hotel Availability



Code (PHP, HTML, JS, CSS, etc.)

Code - Logout

```php
session_reset(); //restart the session, removing any session created data
session_destroy(); //destroys the current session, ending it
header("Location: home.php"); //redirects the user to the homepage
exit(); //terminates the script
```

Code - Login

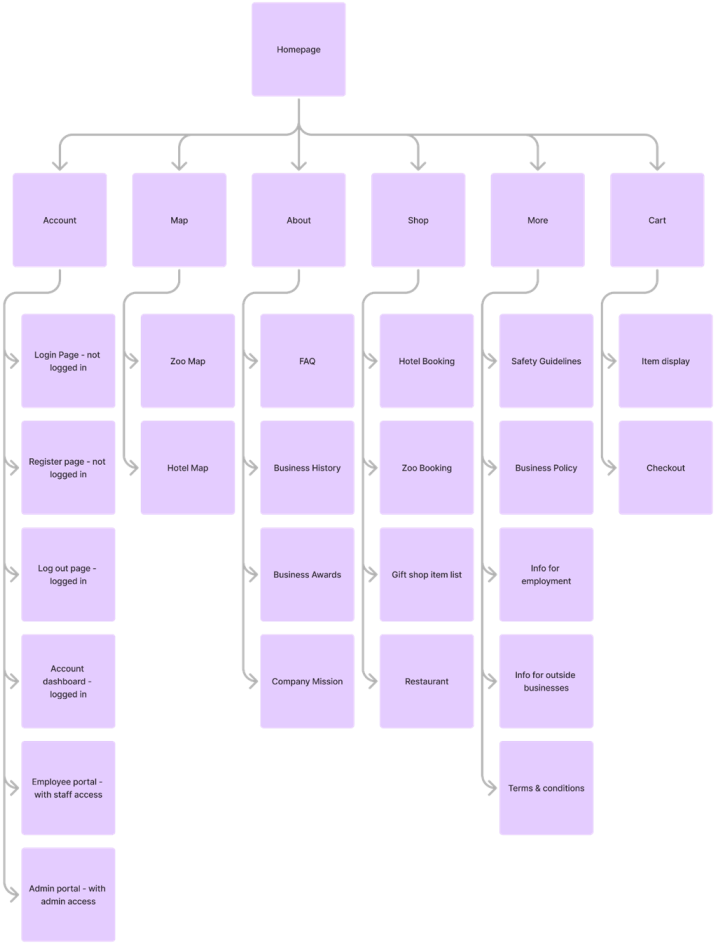
```php
session_start(); //starts a session

if ($_SERVER['REQUEST_METHOD'] == 'POST') { //checks if the form has been submitted
    $email = $_POST['email']; //makes a new variable and sets it to the value of the 'email' input that has been submitted
    $password = $_POST['password']; //makes a new variable and sets it to the value of the 'password' input that has been submitted

    $email = strip_tags($email); //strips the tags in email and password to prevent malicious text from getting in our system
    $password = strip_tags($password);

    $con = new mysqli("localhost", "root", "", "user_information") //creates a new database connection to the user_information database

    if ($con->connect_error) { //checks if the connection to the database has failed/resulted in an error
        die("Connection failed: " . $con->connect_error); //terminates the current script and sends the error in the console
    }

    $stmt = $con->prepare("SELECT password FROM users WHERE email = ?"); //prepares an sql statement to get the password that is in the same row as the email
    $stmt->bind_param("s", $email); //puts the email in the statement
    $stmt->execute(); //executes the statement, runs it through the database
    $stmt->bind_result($hashed_password); //where the result will go, in a variable called 'hashed_password'
    $stmt->fetch(); //aquires the result

    if (password_verify($password, $hashed_password)) { //checks if the password entered matches the hashed password
        $_SESSION['email'] = $email; //tells the session that you are logged in with that email
    } else { //if the password and hashed password dont match
        $error = "Invalid credentials."; //sets an error message to be displayed
    }

    $stmt->close(); //closes the statement
    $con->close(); //closes the connection

    header("Location: dashboard.php"); //sends the user to the dashboard
}
```

Code - Register

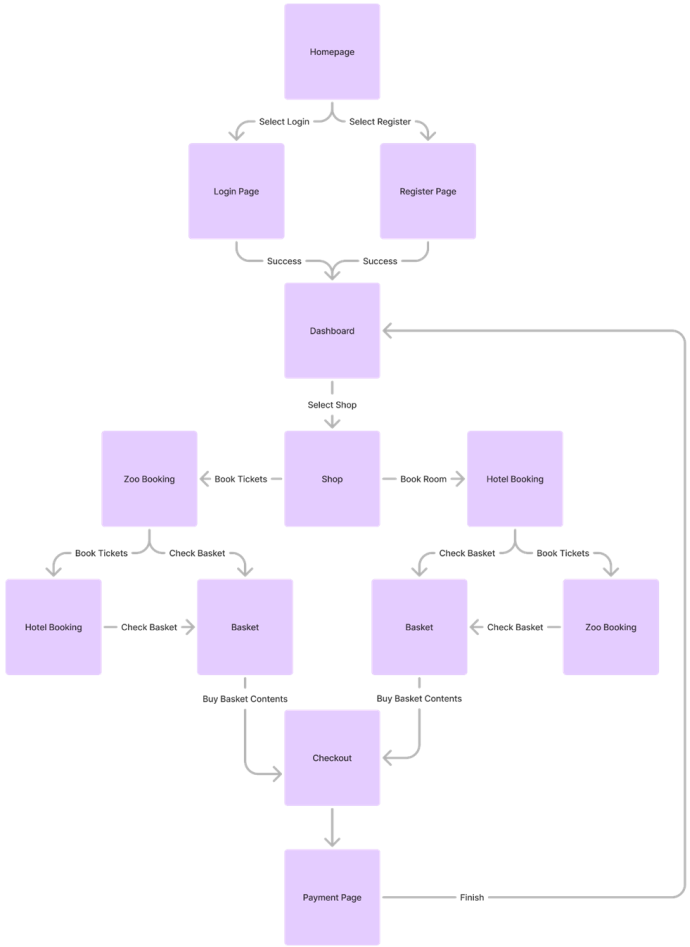
```php
function register() {
    session_start(); //starts a session

    if (isset($_SESSION["email"])) { //checks if the user has logged in already, redirects them if they have
        header("Location: dashboard.php");
    }

    if ($_SERVER["REQUEST_METHOD"] == "POST") { //checks if the form has been submitted
        //gets all the submitted values from super global post, strips any harmful tags and then applies them to the correct variable
        $email = $_POST['new_email'];
        $first_name = strip_tags($_POST['new_firstname']);
        $last_name = strip_tags($_POST['new_lastname']);
        $phone_number = strip_tags($_POST['phoneNumber']);
        $dob = strip_tags($_POST['dob']);
        $password = strip_tags($_POST['new_password']);
        //hashes the password and filters the email to make sure it is an email
        $password = password_hash($password, PASSWORD_BCRYPT);
        $email = filter_var($email, FILTER_SANITIZE_EMAIL);

        $con = new mysqli("localhost", "root", "", "user_management"); //creates a new database connection to the user_information database

        if ($con->connect_error) { //checks if the connection to the database has failed/resulted in an error
            die("Connection failed: " . $con->connect_error); //terminates the current script and sends the error in the console
        }

        try { //trys to insert the values into the table, prevents errors crashing the program
            if (empty($phone_number)) { //checks if phone_number is empty or not, if it is, it excludes it from the sql statement
                $stmt = $con->prepare("INSERT INTO users (email, first_name, last_name, dob, password) VALUES (?, ?, ?, ?, ?)"); //prepares the statement to insert the values into the table
                $stmt->bind_param("sssss", $email, $first_name, $last_name, $dob, $password); //binds all the parameters with the correct values
            }
            else if (empty($dob)) { //checks if dob is empty or not, if it is, it excludes it from the sql statement
                $stmt = $con->prepare("INSERT INTO users (email, first_name, last_name, phone_number, password) VALUES (?, ?, ?, ?, ?)"); //prepares the statement to insert the values into the table
                $stmt->bind_param("sssss", $email, $first_name, $last_name, $phone_number, $password); //binds all the parameters with the correct values
            }
            else { //if both dob and phone_number is not empty, it adds them both to the sql statement
                $stmt = $con->prepare("INSERT INTO users (email, first_name, last_name, phone_number, dob, password) VALUES (?, ?, ?, ?, ?, ?)"); //prepares the statement to insert the values into the table
                $stmt->bind_param("ssssss", $email, $first_name, $last_name, $phone_number, $dob, $password); //binds all the parameters with the correct values
            }

            if ($stmt->execute()) { //executes the statement
                echo "Registration successful!"; //sends confirmation message

                $_SESSION['email'] = $email; //sets the session email to the users email, makes them 'logged in'

                header("Location: dashboard.php"); //redirects the user to the dashboard page

                exit(); //terminates the current script
            }
            else { //if the statement results with an error
                throw new Exception("Duplicate Entry"); //throws an exception to say its resulted in an error
            }
        }

        catch (Exception $e) { //checks if the exception is thrown
            $error = "Email in use"; //sets an error message
        }
        //closes the statement and the connection to the database
        $stmt->close();
        $con->close();
    }
}
```

Site Diagrams

Site Map


Homepage
Account
Map
About
Shop
More
Cart
Login Page - not logged in
Register page - not logged in
Log out page - logged in
Account dashboard - logged in
Employee portal - with staff access
Admin portal - with admin access
Zoo Map
Hotel Map
FAQ
Business History
Business Awards
Company Mission
Hotel Booking
Zoo Booking
Gift shop item list
Restaurant
Safety Guidelines
Business Policy
Info for employment
Info for outside businesses
Terms & conditions
Item display
Checkout

User Flow Diagram



Design

Page Specification

Colour scheme/ why

Font/Font size/ why

Logo / company strapline

How to show interactive elements or input boxes

An overall description of the layout and why you have designed it that way

Accessibility

Visually appealing
Well structured
Including standard layout
Images that reflect stakeholders focus
Accessibility WCAG guidelines
Descriptive alt text
User reviews
FAQs
Security/validation

Wireframes

High, medium, and low fidelity

Visual Designs

Prototypes

Tech Stack

Type	Name	Description
Back End	MySQL 	A database management system, which is easily used and widely known.
Back End	PHP 	A language used for making dynamic web pages and manipulating databases.
Front End	Javascript 	A coding language commonly used for scripting within websites.
Front End	CSS 	A language used to style elements on a web page.
Front End	HTML 	A language used to layout elements on a web page.
API	Stripe 	Stripe is used for collecting payments from users, or initiating subscriptions

Testing

Testing Strategy

Question:

What is the difference between Black Box testing and White Box testing?

We need to consider our overall testing strategy and to document this before we begin the test plan.

Explaining what strategy we used and how we are selecting both what components to test and the data we are testing from them is important.

We also should consider what outcomes from the test we might see and how we are going to respond to them?

Describe Black Box and White box testing

How do they differ in:

Test data

Test criteria

Test outcomes

When will you use this testing?

1. Test the UI:
- Conduct visual testing to ensure elements are laid out correctly.
- Test that all input elements are working correctly.
- Check that all forms are validating data correctly.
- Test navigation across all pages in the application.
- Perform accessibility testing to ensure users with disabilities can access the app.

2. Test the Backend:
- Verify that the APIs are returning the expected responses.
- Test that the data stored in the database is accurate.
- Check that any 3rd party integrations are working as expected.

Test Plans

Task 1: Test strategy

Date of test	Component to be tested	Type of test to be carried out	Prerequisites and dependencies

User Stories

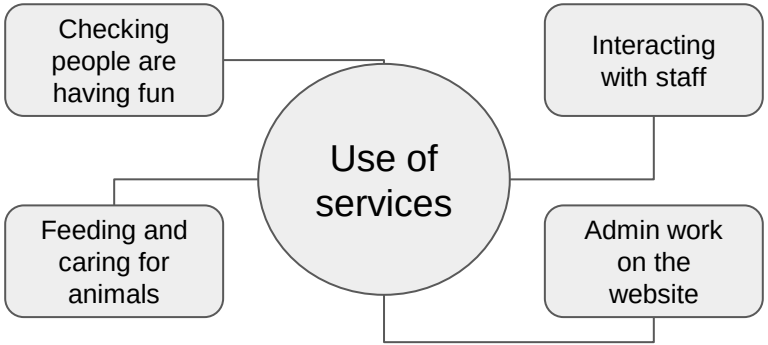
Stakeholder type: Owner
Name: Michael Scott
Age: 35

Characteristics:
- Stern
- Loves animals
- Friendly
- Extrovert
- Rich 🤑

User story:
I am the owner of the zoo, every now and then i go around the zoo and interact with the customers to check they are enjoying themselves.

I also like to feed and care for the animals, so usually some of the staff have a little extra break.

I am fairly lenient with my staff, but I won't have them messing around if stuff still needs to be done.



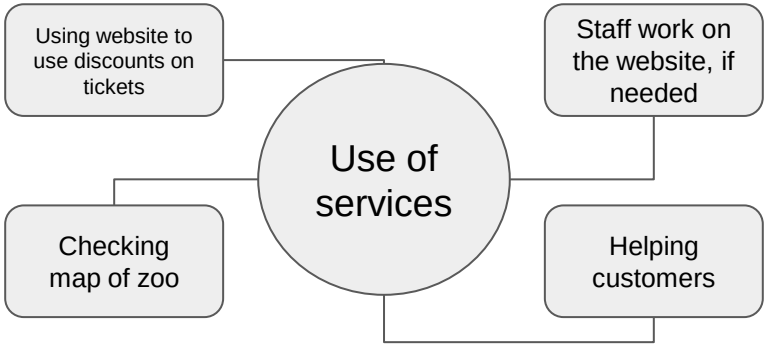
Stakeholder type: Staff
Name: John Smith
Age: 25

Characteristics:
- Confident
- Likes to go out
- Extrovert
- Easy going

User story:
I am working at the zoo, I enjoy the zoo, it’s nice to walk around.

Its large and high quality, so it keeps me occupied.

I am a bit strapped for cash at the moment, it pays well, especially since you get a discount for any services at the zoo.



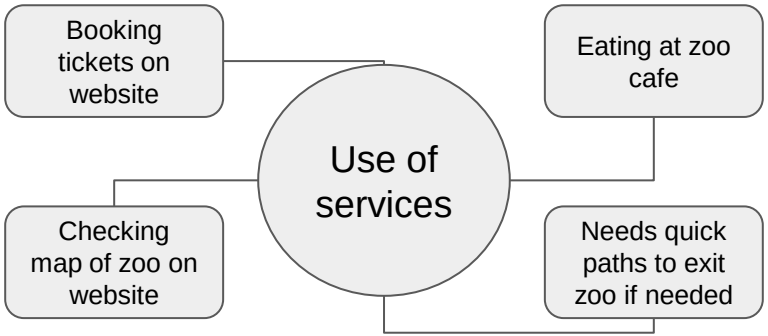
Stakeholder type: Educational
Name: Walter Richard
Age: 69

Characteristics:
- Careful
- Not a risk taker
- Positive

User story:
I am checking out if this zoo is suitable for an educational trip for my students.

I’m hoping this zoo is kid friendly, however I am worried it will be too large and they will get bored too fast.

The cafe should be good quality, I want the kids to have good food if they are buying some.



Stakeholder type: Customer
Name: Michael Scott Jr.
Age: 15

Characteristics:
- Hyperactive
- Introvert
- Reluctant to spend money
- Friendly

User story:
I am going on a trip to the zoo with my friends.

I’m going to get out of cleaning the house, I want the zoo to be big enough to walk around for a while.

I’ve got money for the gift shop, I’m going to buy some fun things.