**Table of Contents**

1. Business Case 3
2. Problem or issue statement 4
3. Revenue model 7
4. Marketing Strategy 8
5. Cost Evaluation 12
6. Functional Requirements 13
7. Non-functional Requirements 15
8. Use Case diagram 16
9. Design Goals 17

10. How to achieve design goals 18

11. Architecture 18

12. Hardware Configuration 26

13. Software Configuration 27

14. Interface Design 29

15. Sequence diagram 35

16. Implementation 36

10.

**Business Case**

**Roaming Razors**

A startup to make grooming easier.

Contact with skilled saloon artists to get home service.

The Roaming Razors Web App aims to provide a convenient platform for customers to book home barber services and for barbers to offer their services. The business case model is built on the following key components:

* **Customer Convenience:**
  + The app leverages the growing trend of on-demand services, allowing customers to book barber services at their preferred time and location.
  + Convenience is enhanced through features like GPS integration, allowing customers to easily set their location and find available barbers nearby.
* **Barber Empowerment:**
  + Barbers can manage their own schedules, indicating their availability and areas they are willing to serve.
  + The platform provides barbers with analytics and insights into customer preferences, helping them tailor their services to meet demand.
* **Service Quality:**
  + The rating and review system not only holds barbers accountable but also acts as a feedback loop for continuous improvement.
  + Barbers with consistently high ratings are highlighted on the platform, increasing their visibility and attracting more customers.

**Problem or issue statement**

**1. Time Savings for Customers:**

* **On-Demand Booking:**
  + The app's on-demand booking system allows customers to schedule barber services at their convenience without the need for advance planning.
  + Customers can find available barbers nearby, eliminating the need to wait in queues or travel to a salon.
* **Efficient Scheduling:**
  + The platform streamlines the booking process, enabling customers to select the desired date, time, and location for the service.
  + This eliminates the time-consuming process of calling or visiting multiple barber shops to find an available slot.
* **Reduced Waiting Time:**
  + By bringing the barber to the customer's location, the app minimizes or eliminates waiting times typically associated with salon visits.
  + Customers can continue with their daily activities until the scheduled time, enhancing overall time management.
* **Flexible Service:**
  + The flexibility to choose a time that suits the customer's schedule contributes to time savings.
  + This is particularly beneficial for busy professionals, parents, or individuals with tight schedules who may find it challenging to allocate specific time slots for salon visits.

**2. Benefits for Barbers Wanting Flex Work:**

* **No Shop Rental Costs:**
  + Barbers who want to engage in flexible work can leverage the platform without the financial burden of renting a physical shop.
  + This allows them to operate without the fixed overhead costs associated with traditional brick-and-mortar salons.
* **Flexible Schedule Management:**
  + Barbers have the autonomy to set their own working hours and availability on the app.
  + This flexibility enables barbers to balance work with personal commitments or pursue other opportunities outside traditional salon hours.
* **Wider Clientele Reach:**
  + The app provides an avenue for barbers to reach a broader clientele, extending beyond the geographic limitations of a physical shop.
  + Barbers can offer services to customers who might not have easy access to a local salon.
* **Increased Income Potential:**
  + The flexibility in scheduling and the ability to serve multiple locations can lead to increased income for barbers.
  + Barbers can optimize their time by accepting appointments in various areas, maximizing their earning potential.
* **Personalized Service:**
  + Providing services in customers' homes allows barbers to offer a more personalized experience.
  + Barbers can build stronger connections with their clients, enhancing customer satisfaction and the likelihood of repeat business.

In summary, the app not only saves time for customers by providing on-demand, convenient services but also empowers barbers by offering a flexible work model that eliminates the need for a traditional shop rental, allowing them to manage their schedules and broaden their clientele reach. The symbiotic relationship between time savings for customers and flexibility for barbers creates a win-win situation for both stakeholders in the home barber service ecosystem.

**\*\* Barbers need to be certified from a recognized institute and need to pass a criminal background check. \*\***

**Revenue Model**

* **Service Fees:**
  + The platform charges a percentage of each transaction (10 %) , ensuring a sustainable revenue stream that scales with the success of the service.
* **Premium Listings:**
  + Premium listings feature barbers at the top of search results, making them more visible to customers. This can be particularly beneficial for new or highly rated barbers.
  + Barbers need to pay a monthly fee of $19.99/month to list their services on Roaming Razors.
* **Subscription Plans:**
  + Subscription plans offer customers benefits such as discounted rates, priority booking, or exclusive promotions. This encourages customer loyalty and provides a predictable revenue source.

**Marketing Strategy**

Creating a successful marketing strategy for Roaming Razors involves a mix of online and offline efforts. Here's a comprehensive marketing strategy:

1. **Digital Presence:**
   * **Social Media Marketing:**
     + Leverage platforms like Instagram, Facebook, and Twitter to showcase before-and-after images, customer testimonials, and promotional content.
     + Engage with the community, run contests, and use targeted ads to reach potential customers and barbers.
   * **Content Marketing:**
     + Create a blog on your website to share grooming tips, trends, and news related to the industry.
     + Develop engaging content that positions your app as a go-to resource for both customers and barbers.
   * **Influencer Partnerships:**
     + Collaborate with local influencers, barbers, and beauty bloggers to promote your app.
     + Influencers can create content, share their experiences with the app, and attract their followers to try the service.
2. **Offline Marketing:**
   * **Local Partnerships:**
     + Partner with local businesses, such as cafes, gyms, or community centers, to display promotional materials or offer exclusive discounts.
     + Establishing partnerships can help you tap into existing local communities.
   * **Flyer Distribution:**
     + Distribute flyers and promotional materials in high-traffic areas or places where your target audience is likely to visit.
     + Include QR codes or unique promo codes to track the effectiveness of your offline efforts.
   * **Community Events:**
     + Sponsor or participate in community events, fairs, or markets to increase brand awareness.
     + Set up booths, offer live demonstrations, and engage with the community.
3. **Referral and Loyalty Programs:**
   * **Referral Discounts:**
     + Implement a referral program where existing customers and barbers can earn discounts or rewards for referring new users to the platform.
   * **Loyalty Rewards:**
     + Introduce a loyalty program that offers perks such as discounted services, exclusive promotions, or early access to new features for repeat customers and active barbers.
4. **Online Advertising:**
   * **Google Ads:**
     + Run targeted Google Ads campaigns to capture users actively searching for home barber services in your target locations.
     + Utilize location-based keywords to enhance local visibility.
   * **Social Media Ads:**
     + Invest in targeted social media advertising to reach specific demographics and geographical areas.
     + Use visually appealing ads and compelling copy to encourage app downloads and sign-ups.
5. **Customer Testimonials and Reviews:**
   * **Showcase Success Stories:**
     + Share success stories and testimonials from satisfied customers and barbers on your website and social media platforms.
     + Encourage users to leave positive reviews on app stores and review websites.
6. **Launch Promotions and Discounts:**
   * **Initial Launch Offers:**
     + Offer special promotions, discounts, or exclusive perks during the initial launch phase to incentivize early adoption.
   * **Seasonal Promotions:**
     + Introduce seasonal promotions or tie-ins with special events to attract attention and drive user engagement.

**Cost Evaluation**

**Setup Costs:**

1. **Development Team:**
   * Hiring a team of developers, designers, and QA engineers may cost anywhere from $12,000 to $15,000, depending on experience levels and project complexity.
2. **Server Infrastructure:**
   * Initial setup costs for cloud hosting services and domain registration might be around $8,000 to $10,000.
3. **Legal and Compliance:**
   * Legal consultation and compliance-related costs might be around $8,000 to $12,000.
4. **Marketing and Launch:**
   * Marketing and launch expenses could range from $20,000 to $30,000, depending on the scale of promotional activities.
5. **Maintenance Cost:**

* The maintenance cost for this web application will cost around $25000-$30000.

The total cost for this web application will be around $100,000.

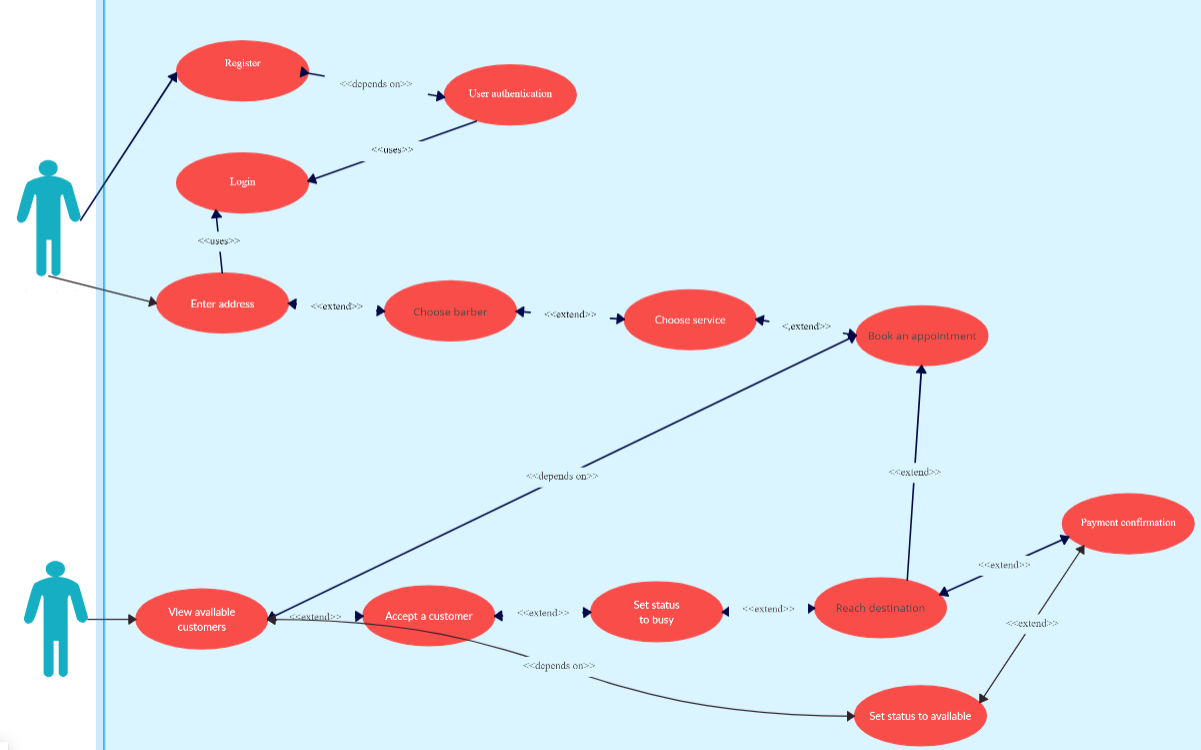
**Functional Requirements**

* **User Authentication:**
  + Two-factor authentication and secure password protocols enhance the security of user accounts, ensuring that personal information remains confidential.
* **Profile Management:**
  + Customers can customize their profiles with haircut preferences, style preferences, and favorite products, providing barbers with valuable insights.
* **Booking System:**
  + The booking system is designed for ease of use, allowing customers to quickly schedule appointments, and providing barbers with clear information about upcoming bookings.
* **Payment Integration:**
  + Multiple payment options, including credit cards and digital wallets, are integrated to accommodate diverse customer preferences and enhance the overall user experience. The app doesn’t accept cryptocurrency and cash as mode of payment currently.
* **Rating and Review System:**
  + The rating system is transparent, with detailed reviews providing valuable information for other users. This ensures a merit-based system and encourages continuous improvement.
* **Real-time Communication:**
  + In-app messaging facilitates quick and direct communication between customers and barbers, enabling them to discuss appointment details or address any concerns.
* **Location-Based search functionality:**
* Location-based search functionality allows the customers to find available barbers near them.

**Non-Functional Requirements**

* **Scalability:**
  + The app is designed with a modular and scalable architecture, capable of handling a surge in users and transactions as the platform grows.
* **Security:**
  + SSL encryption secures data during transmission, and stringent access controls ensure that user data is protected against unauthorized access.
* **Performance:**
  + The app is optimized for fast loading times and smooth transitions between pages. Regular performance monitoring ensures that any issues are promptly addressed.
* **Compatibility:**
  + Cross-browser and cross-device compatibility testing is conducted to guarantee a consistent and enjoyable user experience for all users, regardless of their preferred platform.
* **Data Backup and Recovery:**
  + Automated daily backups ensure that user data is regularly safeguarded. A well-defined recovery plan is in place to minimize downtime in the event of unforeseen issues.

**Use Case Diagram**



**Design Goals**

* **Scalability:**
  + The app is designed with a modular and scalable architecture, capable of handling a surge in users and transactions as the platform grows.
* **Security:**
  + SSL encryption secures data during transmission, and stringent access controls ensure that user data is protected against unauthorized access.
* **Performance:**
  + The app is optimized for fast loading times and smooth transitions between pages. Regular performance monitoring ensures that any issues are promptly addressed.
* **Compatibility:**
  + Cross-browser and cross-device compatibility testing is conducted to guarantee a consistent and enjoyable user experience for all users, regardless of their preferred platform.
* **Data Backup and Recovery:**
  + Automated daily backups ensure that user data is regularly safeguarded. A well-defined recovery plan is in place to minimize downtime in the event of unforeseen issues.

**How To Achieve these Design Goals:**

* **Performance**:

Optimize frontend assets (JavaScript, CSS, images) for faster loading times.

Implement server-side rendering (SSR) or static site generation (SSG) to improve time to first paint and overall perceived performance.

* **Security**:

Implement HTTPS using SSL to encrypt data transmitted between the client and server, ensuring confidentiality and integrity.

* **Scalability**:

Use cloud-based infrastructure provider AWS that offer scalable services like auto-scaling groups, load balancers, and serverless computing.

* **Maintainability**:

Use version control systems (e.g., Git) and collaborative development workflows (e.g., pull requests, code reviews) to manage code changes and ensure code quality.

**Architecture of the system**

1. **Client-side ( Front-end) architecture:**

User Interface: This handles the presentation layer and user interactions. Developed using

1. HTML
2. CSS
3. Bootstrap
4. JavaScript

HTML, CSS, Bootstrap, and JavaScript play essential roles in the development of our web app:

1. **HTML (Hypertext Markup Language)**:

- HTML provides the structure and semantic markup for web pages.

- It defines the layout of elements such as headings, paragraphs, lists, and forms.

- HTML tags organize content and create the basic structure of the web page.

- HTML is used to create the user interface (UI), including forms, buttons, navigation menus, and other interactive elements.

2. **CSS (Cascading Style Sheets)**:

- CSS is used to style and format HTML elements, defining their appearance and layout.

- It allows developers to control properties such as colors, fonts, spacing, borders, and positioning.

- CSS enables consistent styling across multiple pages and ensures a cohesive visual design.

- CSS is used to customize the look and feel of the UI, making it visually appealing and user-friendly.

3. **Bootstrap**:

- Bootstrap is a front-end framework that provides pre-designed CSS and JavaScript components for building responsive web interfaces.

- It includes a grid system, typography, forms, buttons, navigation bars, and other UI components.

- Bootstrap's responsive design ensures that web apps are optimized for various devices and screen sizes, from desktops to smartphones.

- Bootstrap simplifies UI development by providing ready-to-use components that can be easily customized and integrated.

4. **JavaScript**:

- JavaScript is a programming language used to add interactivity and dynamic behavior to web pages.

- It allows developers to manipulate the HTML DOM (Document Object Model), handle user interactions, and respond to events.

- JavaScript enables features such as form validation, animations, dropdown menus, slideshows, and client-side data processing.

- JavaScript enhances user experience by providing interactive features and real-time updates without the need for page reloads.

In summary, HTML defines the structure of web pages, CSS styles them, Bootstrap provides pre-designed components for layout and styling, and JavaScript adds interactivity and dynamic behavior to create a rich user experience in web applications.

1. **Server-Side ( Back-end):**

We're using ASP.NET MVC with C# for our backend, our architecture would be structured around the MVC (Model-View-Controller) pattern. Here's how you can relate these components to our web app:

1**. Model**:

- In ASP.NET MVC, the Model represents the application's data and business logic.

- Models would define the structure of your data entities, such as User, Appointment, Hairdresser, etc.

- Models can also encapsulate business logic related to data validation, calculations, and interactions with the database.

2. **View**:

- Views in ASP.NET MVC represent the user interface layer of your application.

- Views are responsible for rendering the HTML markup that is sent to the client's browser.

- Views also contain JavaScript and CSS for enhancing user interactions and styling.

3**. Controller**:

- Controllers in ASP.NET MVC handle user requests, process input, and determine the appropriate response.

- Controllers receive requests from the client, invoke the necessary operations on the Model layer, and select the appropriate View to render the response.

- Our Controllers would contain action methods that correspond to different routes/endpoints of our API.

- Action methods would handle tasks such as retrieving data from the Model, performing business logic, and returning the appropriate View or data to the client.

4. **Backend Services**:

- In addition to the MVC components, you would have backend services written in C# that handle business logic, data manipulation, and interaction with external services.

- These services would encapsulate complex operations that cannot be directly handled within the Controllers or Models.

- For example, you might have a UserService for managing user authentication and authorization, an AppointmentService for handling appointment scheduling, and a HairdresserService for managing hairdresser profiles and services.

By leveraging ASP.NET MVC with C# for our backend, we can build a robust and scalable web application with a well-defined separation of concerns, making it easier to maintain and extend over time.

**Entity-Relationship Diagram**

**A computer screen shot of a computer program

Description automatically generated**

**Component Diagram**

**A diagram of a software development

Description automatically generated**

**Hardware Configuration**

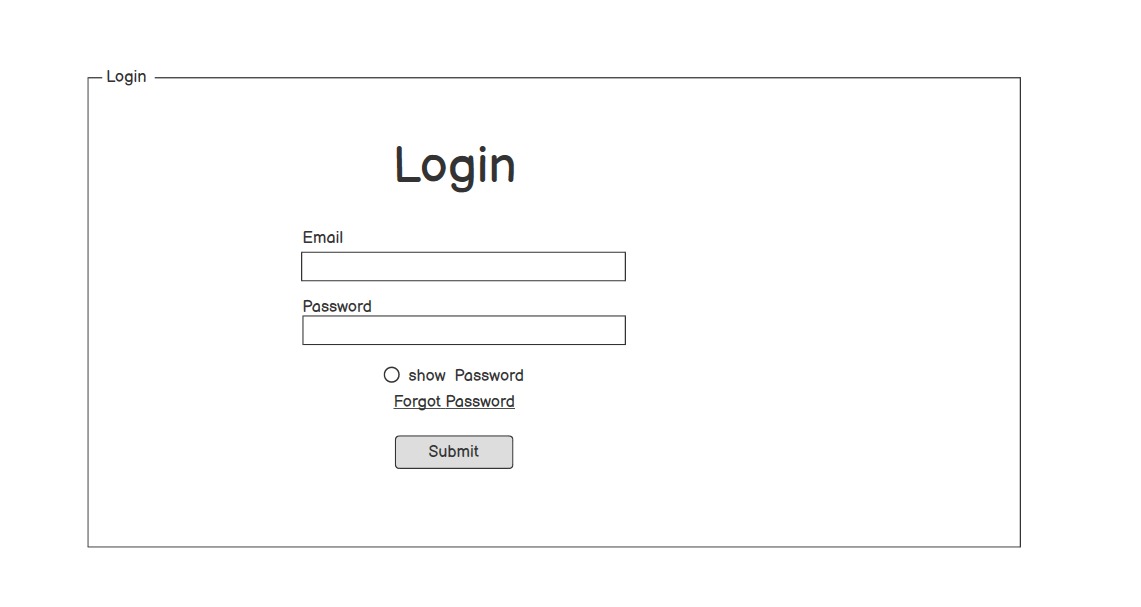
* **Web Servers**: Use servers capable of running the chosen operating system and web server software efficiently. This can include physical servers or virtual machines.
* **Load Balancers**: Implement load balancers to distribute incoming traffic across multiple web servers for improved scalability and reliability.
* **Database Servers**: Deploy dedicated servers or database instances to host your database management system (DBMS), ensuring adequate resources for handling data storage and processing.
* **Storage**: Use reliable and scalable storage solutions for storing static assets, user data, and session information. This can include solid-state drives (SSDs) or cloud storage services.
* **Networking**: Ensure high-speed and reliable network connectivity between servers, users, and external services to minimize latency and ensure smooth data transfer.

**Software Configuration**

* **Database Management System (DBMS)**: Install and configure a relational database management system (RDBMS) SQL Server to store and manage application data.
* **Application Framework**: Set up the necessary software frameworks and libraries for your web application backend, ASP.NET MVC with C#.
* **Development Tools**: Install development tools and utilities for building, testing, and deploying your web application, such as IDEs, version control systems (e.g., Git).
* **Security Software**: Implement security software and practices to protect your web application from common threats, including firewalls, and encryption protocols.
* **Monitoring and Analytics**: Set up monitoring and analytics tools to track the performance, availability, and usage of your web application.

**Interface Design ( Balsamiq wireframes)**

* 1. **Login page**



* 1. **Sign up page**

A screen shot of a sign up form

Description automatically generated

A screen shot of a login form

Description automatically generated

* 1. **Catalogue**

A computer screen shot of a computer

Description automatically generated

* 1. **Services page for customers**

A screenshot of a computer

Description automatically generated

* 1. **Services page for Hairdressers**

A screenshot of a computer

Description automatically generated

* 1. **Admin’s page**

A screenshot of a computer

Description automatically generated

**Sequence Diagram**

A screenshot of a computer screen

Description automatically generated

# Implementation

## Testing

Integration Testing:

* Integration testing is a type of software testing where individual software modules or components are combined and tested as a group. The purpose of integration testing is to ensure that the interactions between these components work as expected and that the integrated system functions correctly.

User Authentication and Authorization Testing: Ensure that the signup process for both customers and freelance hairdressers works correctly. Test that users can sign up, log in, and access their respective accounts securely. Verify that access control mechanisms are properly implemented to restrict unauthorized access to certain features or data.

Payment Processing Testing: Test the payment processing functionality for handling transactions between customers and hairdressers. Verify that customers can make payments for services rendered, and that hairdressers can receive payments securely. Test different payment methods (e.g., credit card, PayPal) and ensure that transactions are processed smoothly.

|  |  |
| --- | --- |
| Test Case | Test Result |
| User Authentication | Pass |
| Payment Processing | Pass |

Security Testing:

* Security testing plays a crucial role in ensuring the integrity, confidentiality, and availability of information within a system. By proactively identifying and addressing security vulnerabilities, we can mitigate risks and protect our assets from potential security threats.

Authentication and Authorization Testing:

Verify that the authentication mechanisms, such as username/password authentication are implemented securely to prevent unauthorized access.

Test for proper session management to ensure that sessions are securely managed and invalidated after logout or timeout.

Buffer Overflow:

Test for buffer overflow vulnerabilities in user input fields, particularly in areas where user-provided data is processed or stored, such as login forms, search bars, and order details. Buffer overflow vulnerability detected in the order details section where the application fails to properly validate user input before processing.

|  |  |
| --- | --- |
| Test Case | Test Result |
| Authentication | Pass |
| Buffer Overflow | Fail |

Regression Testing:

* Regression testing plays a crucial role in maintaining the stability, reliability, and quality of software applications by ensuring that they continue to perform as expected throughout the software development lifecycle, despite ongoing changes and updates. It helps detect and prevent regression issues early in the development process, minimizing the risk of introducing defects into production environments.

Authentication and Authorization Regression Testing:

Verify that users can still sign up, log in, and access their accounts securely after any changes to the authentication and authorization mechanisms. Ensure that access control measures are functioning correctly to prevent unauthorized access to sensitive features and data.

Payment Processing Regression Testing:

Verify that users can still make payments for services rendered, and that hairdressers can receive payments securely without any issues. Test different payment methods (e.g., credit card, PayPal) to ensure that transactions are processed smoothly.

|  |  |
| --- | --- |
| Test Case | Test Result |
| Authentication | Pass |
| Payment Processing | Pass |