Text

Description automatically generated

**Faculty of Science**

**Department of Mathematics and Computer Science**

**Senior Project - II (CMPS 444)**

**[GO]**

**Student Name: Student ID:**

[Youssef Hoteit] [202303580]

[Omar Arnaout] [202202633]

[Mohamad Abdulhadi] [202303164]

[Hady Hamoud] [202301058]

**Supervisor:**

[Dr. Bilal Nakhal] Principal Supervisor

**Acknowledgements**

I'd want to thank everyone who helped make Go: A Local Tourism Platform in Lebanon a success. This initiative would not have been feasible without the joint efforts of our devoted team members, whose enthusiasm for promoting tourism in Lebanon has driven the development of this unique platform.

Special thanks go to the local companies and tourist experts whose insightful views and participation helped us shape Go to meet the market's specific demands. Your dedication to highlighting Lebanon's beauty and variety has helped us achieve our goals.

I am also grateful to the technical professionals that created and implemented Go's advanced AI technology and responsive design. Your knowledge has helped us turn our idea into a reality that will set the standard in travel technology.

Furthermore, we thank the academic and cultural consultants that expanded our platform with their expertise, ensuring that GO remains a real representation of Lebanon's rich past.  
Our financial supporters and investors' conviction in Go's potential has been critical. Your contribution has helped us to strive for and achieve excellence in our activities.

Finally, I'd want to thank the user community for their continuous support of our platform. Your feedback and excitement drive us to develop and innovate.   
We worked together to establish a platform that not only allows travelers to explore but also helps local economies and encourages sustainable tourism practices. Thank you all for helping to establish Go as a cornerstone of Lebanon's tourist industry.

With most gratitude and warm regards,

Go’s Team

**Abstract**

Go is an innovative website that aims to develop Lebanon’s tourist industry by the power of intelligence to create individualized travel experiences. This website compiles thorough, up-to-date information about Lebanon's cultural, historical, and natural landmarks into a single database. Go uses an AI-driven UI to provide bespoke suggestions and itineraries that cater to the tastes of both local and foreign consumers. Key features include interactive maps, multimedia information, and real-time updates on local events, all available in different languages such as Arabic, English.

The algorithm constantly monitors user interactions to optimize suggestions, resulting in a relevant and engaging user experience. This method not only facilitates navigation among Lebanon's numerous tourist attractions, but it also promotes lesser-known places, resulting in more equally distributed tourism traffic and sustainable tourism practices. Go gives local companies more visibility and direct access to potential clients, which promotes economic growth in the neighborhood.   
Go strives to be the ultimate platform for travelers in Lebanon by combining innovative technology with a focus on user-centric design, boosting discovery and enriching travel experiences while encouraging local business growth and sustainable development in the tourism sector.

**Table of Contents**

[**Chapter 1. Introduction** 6](#_Toc135824237)

[**A.** **Main Project Description** 6](#_Toc135824238)

[**B.** **Problem Statement** 6](#_Toc135824239)

[**C.** **Project Significance and Goal** 6](#_Toc135824240)

[**D.** **System and Domain Review** 6](#_Toc135824241)

[**Chapter 2. Project Management** 7](#_Toc135824242)

[**A.** **SDLC Model** 7](#_Toc135824243)

[**B.** **Risk Management** 7](#_Toc135824244)

[**C.** **Project Organization** 7](#_Toc135824245)

[**D.** **Schedule/Timeline** 7](#_Toc135824246)

[**E.** **Project Management Tools Used** 7](#_Toc135824247)

[**Chapter 3. Software Requirement Specification** 8](#_Toc135824248)

[**A.** **Feasibility Study** 8](#_Toc135824249)

[**B.** **Requirement Elicitation** 8](#_Toc135824250)

[**C.** **Product Functions** 8](#_Toc135824251)

[**D.** **User characteristics** 8](#_Toc135824252)

[**E.** **Non-functional Requirements** 8](#_Toc135824253)

[**F.** **Domain Requirements** 8](#_Toc135824254)

[**G.** **Functional Requirements Specification** 9](#_Toc135824255)

[**Chapter 4. Project Design** 10](#_Toc135824256)

[**A.** **User Interface Prototype** 10](#_Toc135824257)

[**B.** **Data Flow Diagram** 10](#_Toc135824258)

[**C.** **Use Case Diagram** 10](#_Toc135824259)

[**D.** **Database Diagram** 10](#_Toc135824260)

[**E.** **Domain/Class/Object Diagrams** 10](#_Toc135824261)

[**F.** **Sequence Diagrams** 10](#_Toc135824262)

[**Chapter 5. Implementation** 11](#_Toc135824263)

[**A.** **Software Architecture** 11](#_Toc135824264)

[**B.** **Technology Stack** 11](#_Toc135824265)

[**C.** **Front-End Implementation** 11](#_Toc135824266)

[**D.** **Back-End Implementation** 11](#_Toc135824267)

[**E.** **Deployment and Delivery** 11](#_Toc135824268)

[**F.** **CS Knowledge Application and Learning** 11](#_Toc135824269)

[**Chapter 6. Testing and Maintenance** 12](#_Toc135824270)

[**A.** **Testing** 12](#_Toc135824271)

[**B.** **Maintenance** 12](#_Toc135824272)

[**Chapter 7. Conclusion and Future Work** 13](#_Toc135824273)

[**Chapter 8. References** 14](#_Toc135824274)

[**Appendices** 15](#_Toc135824275)

# **Chapter 1. Introduction**

## **Main Project Description**

Oftentimes people miss out on most of the opportunities to discover new cultures in different countries. This is exactly where Go app, comes in as it allows users to not only visit different sites in Lebanon but also find deep hidden gems and important events throughout the country. It is a great tool for both a traveler who seeks to explore new regions and a local who wants to learn more about their homeland.

Experience everything in Lebanon has to offer with Go, it is easy to use the app thanks to its very simple interface. You can not only find information about the cultural impact of Lebanon and its history but also find out how many beautiful places are located throughout the country. Instead of focusing solely on the sites you can learn more and explore Lebanon thanks to this application.

This application is perfect for people who have a specific appreciation for different topics such as arts, history and more. The application allows you to adjust the type of content you want the app to provide you with so you will be able to enjoy many unique activities and entertainments throughout your trip.

Say goodbye to the time wasting for planning. Go integrates powerful features like ai chatbot, map for locations, event ticket purchasing, and bookings directly through the app.

You don’t have to worry about staying behind. This app sends notifications regarding ongoing or recently started miniatures and other events. Whether it is a new art installation, a local food festival, or any other event, you will know about it and its particulars instantly.

Stay in Touch, Go has also a social feature where users get to share their experiences and advice with fellow voyagers. Publish your findings, review your excursions and other users who have similar interests.

When the Go app is being refined, along with new versions there’s potential of new features such as a community of local people offering their tours and suggestions in conjunction with AR guided expeditions showcasing the rich past of Lebanon through a touch of your device.

Embark on an adventure with Go and reimagine how you experience everything Lebanon has to offer. It’s not about the relocations that they make only, but the experiences and friendships that they built. Go is your gateway to the best of Lebanon, making sure that each of the trip is as educational as it is entertaining.

## **Problem Statement**

When attempting to plan and manage their journeys as effectively as possible, visitors encounter a variety of challenges in the fast-paced world of travel and tourism. The consequences, beyond the personal loss for those who are unable to experience a destination like a local, are numerous: Well-known tourist destinations become increasingly crowded with tourists, and tourists spend countless hours planning an itinerary that they believe will show them everything about a particular place – time that is far better spent experiencing it! This level of preparation often results in an overtly rigid schedule that does not permit travelers to venture off the beaten track and discover some hidden gems or secret favorite local spots along the way. Those that follow exact replicas of backwards-walking tour guides ignore lesser-known locations that are just as deserving of being on everyone's bucket list. Even worse in nations with rich histories and dynamic cultures, such as Lebanon, is the lack of access to the right technology that enable individuals to discover a city's secrets without having to conduct months or weeks of research before visiting.

Lebanon's tourism industry has two challenges: many undiscovered gems go undiscovered while popular places are overrun with tourists. In addition to degrading the quality of the visitor experience, this mismatch places excessive strain on local resources and infrastructure. However, local festivals, tiny enterprises, and artisan stores that may provide tourists with a genuine look into Lebanon's rich culture and legacy are frequently overlooked.

Our Vision: Go

The goal of Go is to transform how people view Lebanon. This cutting-edge app will serve as a personal travel companion, assisting both residents and visitors in exploring the nation in sustainable and customized ways. Go aims to encourage travelers to venture beyond the typical tourist paths and discover undiscovered gems by fusing state-of-the-art technology with a profound appreciation for Lebanon's culture.

Go strives to provide a sustainable, inclusive, and well-balanced travel experience:• Improving Accessibility: To lessen dependency on congested hotspots, promote lesser-known locations and make it easier to visit a variety of activities.• Promoting tourism: distributing visitors equally to lessen the strain on popular destinations while highlighting the distinct beauty of smaller, locally owned attractions.• Strengthening Local Economies: Assisting small enterprises by incorporating them into the tourism ecosystem and making sure that each trip directly supports local livelihoods.• Travel Planning: Ensuring every trip is smooth and optimized by providing customized itineraries that adjust to current circumstances, such as local events or weather.   
  
Go aims to provide a tourism experience that benefits all parties involved, including tourists seeking educational experiences and locals eager to share their customs and stories.

## **Project Significance and Goal**

The major purpose of our Go initiative is to offer a unified, user-friendly platform that redefines how both visitors and locals connect with Lebanon’s vast cultural richness, stunning landscapes, and vibrant public life. In a country where beauty and history lie around every corner—yet often go unnoticed due to a lack of accessible information—Go steps in to bridge that gap. Whether someone is wandering the old souks of Sidon, craving a mountain hike in Ehden, or curious about the next village food festival, Go helps them discover it all.

Lebanon's tourism industry has always been a sleeping giant—full of potential, but underutilized due to fragmented information and limited global exposure. A well-designed tourism platform like Go has the power to unlock that potential. It’s more than just a convenience tool; it’s a gateway for reviving the local economy, creating jobs, and showing the world a side of Lebanon that's often overshadowed.

Tourism has a real, measurable impact. When more people visit Lebanon, they stay in local guesthouses, eat at small family-owned restaurants, and buy crafts made by local artisans. Every visit supports farmers, guides, drivers, chefs, and countless others whose livelihoods depend on a steady flow of visitors. With Go, we're not just promoting places—we're investing in people.

The app will actively guide users toward experiences that matter—cultural festivals, artisan fairs, nature reserves, and historical sites—by providing updated, engaging content. This includes everything from curated travel paths and personalized suggestions to expert-led virtual tours and real-time event updates. Whether someone has three days or three weeks, Go helps them experience a well-balanced journey that reflects Lebanon’s diversity.

In addition to making planning easier, Go is designed to spread tourism more evenly across the country. Instead of overcrowding a few “famous” spots, it highlights lesser-known villages, hidden beaches, and local events that deserve recognition. This ensures that tourism benefits are shared more equitably, supporting communities that are often left out of traditional travel itineraries.

At its heart, Go is about more than just logistics, it’s about storytelling. By integrating photos, videos, expert talks, and stories from everyday people, the platform helps users truly understand the spirit behind the places they visit. It transforms a random detour into a meaningful experience, and a simple tourist stop into a moment of learning and appreciation.

## **System and Domain Review**

**System Review**

*Go* is built with one simple but powerful idea in mind: making travel in Lebanon easier, richer, and more personal. The app is designed as a full-service tourism companion that gives users everything they need at their fingertips—from finding destinations to planning full experiences based on their own interests.

The home screen acts as a welcoming gateway, neatly organizing all the different categories of destinations—whether someone is looking for historical sites, beach resorts, hiking trails, or hidden village festivals. Instead of overwhelming users with endless options, *Go* provides a clean, intuitive design that helps them navigate smoothly and find exactly what they’re looking for.

One of the standout features of *Go* is the AI chatbot. Visitors can ask questions about destinations, events, transportation, or even cultural tips, and the chatbot provides real-time, human-like assistance. It’s like having a local friend in your pocket, ready to guide you through every step of your journey.

The map page is another core part of the experience, offering users a dynamic way to visually explore where places are located. Whether planning a trip across different regions or just checking what's nearby, the map helps make exploration effortless and more adventurous.

In addition, users can book activities, read and leave reviews, save favorite destinations for future trips, and receive personalized recommendations based on their search history and stated preferences. The app keeps learning from the user’s behavior, becoming smarter and more helpful the more it’s used.

To give users a fully tailored experience, *Go* also features user accounts where travelers can manage their bookings, wishlist, and personal information. Beyond that, there are business accounts as well, allowing local businesses—whether guesthouses, tour operators, or restaurants—to showcase their services, promote special offers, and engage directly with visitors. This creates a healthy ecosystem where tourism grows on both sides: the traveler finds authentic experiences, and local businesses benefit from greater visibility.

In short, *Go* isn’t just another travel app; it’s a living platform that adapts to each user and aims to make discovering Lebanon both effortless and unforgettable.

**Domain Review**

Tourism in Lebanon has always held enormous promise. With its unique blend of ancient history, stunning geography, vibrant culture, and world-renowned hospitality, Lebanon has everything a traveler could wish for. Yet, despite this richness, the tourism sector has often struggled—partly due to the lack of modern tools that simplify planning and bring hidden gems into the spotlight.

This is where *Go* steps in. The app is part of a new wave of digital solutions designed specifically for Lebanon’s tourism domain, focusing on accessibility, personalization, and economic empowerment.

Historically, tourists visiting Lebanon had to rely on scattered blogs, outdated websites, or word-of-mouth advice to plan their trips. There was no centralized, trusted space that brought together all types of travel experiences—from exploring Tyre’s ruins to hiking the Qadisha Valley or discovering underground art scenes in Beirut. *Go* fills that gap by offering an all-in-one solution that respects both the tourists’ need for ease and the local community’s need for support.

By allowing users to connect directly with local businesses, *Go* helps decentralize tourism income, encouraging visitors to venture beyond a few popular spots and into lesser-known towns and villages. This broader exposure doesn’t just enhance travelers’ experiences—it directly stimulates the Lebanese economy at a grassroots level, where every visitor counts.

Moreover, *Go* aligns perfectly with global trends toward more personalized, experience-driven travel. Today's travelers aren’t looking for generic tours; they want meaningful, customizable adventures that match their passions. With features like AI recommendations, dynamic mapping, and personalized itineraries, *Go* responds to this new way of exploring the world.

In a country rebuilding its global image and strengthening its economy, a platform like *Go* is not just useful—it’s essential. It opens a window for tourists to fall in love with Lebanon’s many faces, and it helps ensure that tourism can once again become a sustainable pillar of national growth.

# **Chapter 2. Project Management**

## **SDLC Model**

To successfully handle the dynamic and intricate needs of both tourists and local companies in Lebanon, we decided to use the Agile Scrum approach for the Go app's development. We can handle the many demands of developing a comprehensive tourist platform with the help of a committed development team that includes Omar, Hadi, Youssef, and Mohamad, and the knowledgeable direction of our supervisor, Dr. Bilal Nakhal..Agile Scrum is especially well suited to our needs. It enables us to divide the enormous project into manageable sprints, which usually run one to two weeks. Dr. Bilal assists in defining sprint goals during the planning meeting at the beginning of each sprint, which the team subsequently carries out through daily chores and ongoing testing. Regular stand-up meetings ensure that issues are swiftly fixed, and that progress is maintained. Every sprint ends with a review to present the work to stakeholders and solicit their feedback, followed by a retrospective meeting to refine our process.

This cycle ensures thorough testing and validation of all capabilities, allowing for rapid development and response to changing requirements. Dr. Bilal oversees the deployment of a minimum viable product or complete feature set to ensure stability and operation. The app is periodically updated to reflect user feedback and new requirements, ensuring its continued usefulness after launch. This meticulous yet adjustable technique ensures that GO exceeds consumers' expectations, leading to a more fulfilling trip to Lebanon.

## **Risk Management**

Building Go is an exciting journey, but like any ambitious project, it comes with its share of risks. Part of what makes a strong project successful is not ignoring these risks—but facing them head-on, planning for them early, and adapting as challenges arise. In developing Go, we’ve thought carefully about potential obstacles and how to manage them to keep the app—and the user experience—as strong and smooth as possible.

1. **Technical Risks:**

**Risk:**  
There’s always the chance of technical issues, whether that’s bugs at launch, app crashes, slow performance, or integration problems (especially with live features like booking systems, maps, and the AI chatbot).

**How We Manage It:**  
We’re putting a big focus on **thorough testing**—from early prototypes to final versions—so we can catch issues before users do. We’re also choosing **reliable, scalable technologies** from the start, and building a strong backend that can handle growth without glitches. After launch, a **dedicated technical support** team will monitor performance and react quickly to fix any problems.

1. **Security and Privacy Risks**

**Risk:**  
Since users will be creating accounts, saving destinations, and possibly making bookings, protecting their personal data is critical. Any breach of trust could seriously damage Go’s reputation.

**How We Manage It:**  
We are committed to following **best practices in cybersecurity**—including encryption of sensitive data, regular security audits, and compliance with international data protection standards like GDPR. Users will also have **full control over their data**, with clear privacy settings and options to manage or delete their information whenever they want.

1. **Financial and Operational Risks:**

**Risk:**  
Every project faces financial pressure, whether from unexpected development costs, lower-than-expected revenue, or economic changes that affect user spending.

**How We Manage It:**  
We’re building Go with **cost control and scalability** in mind—starting lean, testing carefully, and expanding features based on real user demand rather than guesswork. Revenue streams will be diversified through bookings, premium features, business partnerships, and potential advertising—keeping the app financially healthy while staying true to the user experience.

In short, we know there are risks—but we also know that with the right preparation, flexibility, and constant listening to our users, we can turn those risks into opportunities for improvement. *Go* isn’t just about building an app; it’s about building trust, excitement, and a community around discovering the best of Lebanon.

## **Project Organization**

When we first kicked off the development of Go, we knew that success would depend on how well we played to each team member’s strengths. So we sat down together, had a few honest conversations, and made sure everyone was taking on a role where they could truly shine. That early planning set the tone for everything that followed.

Youssef took the lead on designing the user experience and interface. His focus was on making Go feel natural and easy to use—from the way users browse destinations, to how they interact with the map and chat with the AI assistant. Youssef worked carefully to strike a balance between beauty and simplicity, making sure that everything looked clean while still feeling familiar and effortless. His goal was always to keep the spirit of adventure alive in the app without overwhelming users.

Omar was the brain behind the database structure and models. Knowing that Go would handle everything from user accounts and business profiles to bookings and reviews, he made sure the system was built for speed, clarity, and growth. Thanks to his thoughtful planning, the database made the rest of the development work much smoother and more efficient.

When it came to the backend development, Omar and Youssef teamed up to bring Go’s core functionalities to life. They tackled everything behind the scenes—from user authentication to booking management, from building the recommendation system to powering the chatbot. Their collaboration made sure that while the app looks simple and seamless on the outside, it’s backed by a strong, reliable engine underneath.

Finally, Hady and Mohamad put their energy into building the frontend of the app. They worked closely with the designs, translating each screen, button, map view, and form into a smooth, responsive experience. Their attention to detail made sure that Go feels fast, intuitive, and ready for users across all types of devices.

From start to finish, every piece of Go came together through teamwork, communication, and a shared vision of creating more than just another app—we built a travel companion that feels like it belongs in the hands of anyone looking to explore the heart of Lebanon

## **Schedule/Timeline**

In order to distribute the tasks among the team members and prevent any possible delay of the project, we created a weekly schedule all through the fall semester to ensure quality and fast productivity are met. We assigned the tasks as follows:

|  |  |  |
| --- | --- | --- |
| Time in weeks | task | Team Members |
| Week 2 | Generate the idea | All |
| Week 2 | Competitive market research | Mohamad/Omar |
| Week 2 | Project Proposal | - |
| Week 3 | Discuss the idea | all |
| Week 3 | Project description | Omar |
| Week 3 | Problem statement | Omar |
| Week 4 | Project goal | hadi |
| Week 4 | System and domain review | Mohamad |
| Week 4 | SDLC Model | Youssef |
| Week 4 | Project Organization | Hadi |
| Week 4 | Ethical standards | Omar |
| Week 5 | Timeline | Mohamad |
| Week 6 | Feasibility study | Omar |
| Week 6 | Project Plan report | all |
| Week 6 | Present to the committee | Mohamad |
| Week 6 | Software requirements | Omar/Youssef |
| Week 7 | Use Case Diagram | Hadi/ Mohamad |
| Week 7 | Porotype | Youusef/Omar |
| Week 7 | DFD | Mohamad |
| Week 7 | Database Diagram | Omar/Mohamad/hadi |
| Week 7 | Uml Diagram | Hadi |
| Week 7 | Sequence | Mohamad/Hadi |
| Week 8 | Present to committee | all |
| Week 10 | Software architecture | Omar and Mohamad |
| Week 11 | UI prototype | Youssef |
| Week 12 | Methodology | Hady |
| Week 13 | Final report draft | All |
| Week 14 | Final report | All |
| Week 15 | Present to comitee | All |

## **Ethical standards and guidelines**

Every successful company is built on ethics, and Go must adhere to the highest standards of ethics in all facets of our corporate operations. We pledge to always act in the community's best interests, to treat everyone fairly and with support, to embrace lifelong learning, and to always conduct ourselves in a responsible and moral manner.

Go will Thoroughly safeguard user privacy and guarantee the privacy of their information. Before using a user's camera for any app functionality, we obtain their express consent. We only show relevant and user-appropriate ads. The location of users will not be accessed unless it is required for a future update or app development.

# **Chapter 3. Software Requirement Specification**

## **Feasibility Study**

Determining the viability of the suggested Go

system in the current economic climate is the main goal of this feasibility study.   
The study's findings will give us the foundation for deciding whether to move forward with the development, modify the system to better fit the current economic climate, or, if the project is not viable, to scrap it altogether. The operational, technical, and financial facets of the project that require careful examination will all be covered in our analysis.   
Risk management involves identifying and reducing any potential obstacles that could prevent a project from being completed successfully. This targets unforeseen challenges that could impact people, procedures, technology, and assets. Our approach to risk management in the Go application would entail classification.

1. **Project Risks:**

• Scope creep: Because there were so many features and concepts at the beginning of the Go project, scope creep was a risk. To combat this, we set clear project parameters at the outset, discussing and deciding on the project's scope with all parties involved. This clarity helped to decrease the possibility of scope extension. We also produced frequent progress updates to ensure that the project kept within its original parameters.

• Performance Risks: Go had performance issues during development, distinctly slow load times, which can bother users. After discovering that the time it took to load images was the root cause, we created a caching solution that stores images locally, significantly reducing load times and boosting overall app performance.

•Schedule Risks: Also known as project schedule risks, there are the likelihood that project activities may take longer than expected, affecting the project's budget, delivery date, and overall performance. To mitigate this risk, we conservatively estimated the time required for each feature and incorporated buffer durations to cater for any unexpected delays. Using this method allowed us to better control timelines and avoid delays.

•Frequent Data Backups: To ensure the secure preservation of critical data, such as user profiles, preferences, and transaction histories, we have devised a strict schedule of frequent data backups.

• Redundancy Measures: If one storage system fails, additional copies of the data are retained to prevent full loss. These copies are kept separately in different media for storage.

• Multiple Storage Locations: We save backup data on cloud storage systems, remote servers, and external hard drives. Because of this layered approach to data storage, data recovery may be completed quickly and effectively, if necessary, while simultaneously improving data security and accessibility.

By employing these measures, we want to protect the Go app against data loss while still providing our users with ongoing, dependable service.

1. **Operational Feasibility**

Operational viability assesses how effectively a proposed solution would resolve an issue once implemented. An online study that looked at the need for better tourism management in Lebanon discovered that there is a significant demand for services that help visitors and residents identify and experience the country's attractions. The survey findings revealed several specific difficulties: Many visitors are unaware of lesser-known destinations and cultural events, resulting in missed opportunities for fascinating experiences; travelers typically feel overwhelmed by the range of options and lack the ability to properly prioritize or manage their trips. Authentic local experiences are becoming increasingly popular, yet tourists sometimes fail to identify or relate to these offers.

Given Lebanon's present financial situation, promoting internal tourism to help in prosperity is becoming increasingly popular. The demand for a service like Go is growing as families and people seek economically and exciting ways to travel inside their own country. In addition to assisting users in finding and planning excursions to well-known and lesser-known destinations, this app will allow users to organize their travels based on their own interests and current information.

Furthermore, there is a growing need for a service that supports sustainable and responsible tourism as more people become aware of its advantages. With features that encourage finding local companies, craftspeople, and environmentally responsible activities, Go will be a great fit for current trend toward mindful travel.

Therefore, Go will serve as an essential tool for enhancing the tourist experience in Lebanon, supporting local economies, and promoting a more connected and sustainable approach to exploring the nation’s rich cultural and natural heritage.

1. **. Technical Feasibility:**

Technical viability evaluates the hardware and software needed to successfully execute a concept and satisfy user requirements. The Go app is technically possible since it makes use of readily accessible software tools and current technologies. To ensure a reliable and scalable application, this comprises a variety of programming languages and development frameworks appropriate for both online and mobile platform development.

**Required Hardware:**

• Individual laptop computers for testing and development.   
  
**Essential Software:**

• Development Tools: We employ cutting-edge editors, libraries, and programming languages like Node.js for backend services, React for online applications, and Flutter for

mobile development.

• Design Tools: To better imagine the user experience and interface prior to actual development, user interface designs are created using programs like Sketch and Adobe XD

.• Presentation and Documentation Tools: We use Microsoft Word to keep clear and comprehensive reports of the development process, and we utilize Microsoft PowerPoint to present project plans and progress.

**Human Resources:**

•A team of at least four skilled developers with experience in cross-platform development, capable of handling the frontend, backend, and database management to bring the Go application to life

1. **Economic Feasibility**

An important study that assesses our project's prospective revenue and cost-effectiveness is economic feasibility. Because of its modest initial development expenses, the Go app is expected to be financially viable. To minimize our upfront expenses, we want to use free open-source tools and frameworks for development. However, once we go public, we will incur extra costs, such hosting fees for the website and app.

**Calculating the User Base and Profit Potential:**

• Market Size: The mobile user base in Lebanon is quite active. We could target about 429,000 potential users out of the 4.29 million mobile users, if even a small percentage, say 10%, use our app because they are interested in local tourism. Our prospective user base is further increased by considering travelers who travel to Lebanon

.• Global Reach: Go focuses on both locals traveling back home and tourists curious about Lebanese culture. Its features, however, can be modified for use in different areas, which could increase our market's size globally.

**Budgeting and Costs:**

•Server/Cloud Server: $400 - $800 for robust server support to handle user traffic.

•App Store and Play Store Fees: $125 for publishing the app.

•Development Tools: Free or low-cost due to our choice of open-source software.

•Website Domain Name: $20 per year.

•Operational Costs: Includes marketing, maintenance, and customer support estimate

d at between $150 - $300.

•Total Estimated Cost: Ranges from $1,200 to $1,500, averaging around $1,350 for initial setup and launch.

**Revenue Streams:**

•In-App Advertising: Projected to generate between $10 to $200 per day, assuming moderate user engagement.

•In-App Purchases and Subscriptions: Offering premium content such as exclusive travel guides, downloadable content, and ad-free experiences could further enhance revenue.

•Estimated Monthly Revenue: With strategic marketing and steady user growth, revenue could quickly scale from $4,000 with 1,000 active users to $20,000 with 5,000 active users per month.

**Break-Even Analysis:**

•We estimate reaching our break-even point within the first eight months with a steady increase in user adoption, requiring around 400 active users to cover ongoing operational costs.

1. **Delivery:**

Go will offer thorough User Documentation for user assistance with the goal of improving user comprehension and app engagement:• functions of the user documentation: these will contain comprehensive explanations of the app's functions, troubleshooting advice, user safety rules, and a quick start guide for instant usability.• Accessibility: The documentation shall be clear and easy to use, enabling users to confidently and independently explore all of the app's capabilities.   
  
Go is not only financially feasible but also well-positioned for expansion and success in boosting Lebanese travel thanks to its meticulous planning.

## **Requirement Elicitation**

Our application GO makes exploring, saving, and booking local travel experiences as easy as sending a text—no special skills required. From the moment you open GO, our streamlined, mobile-first interface guides you through every step:

**Discover & Filter**

• Browse hundreds of destinations instantly by category (restaurants, hotels, landmarks, outdoor activities, and more) or by district

• Use our search bar and filters to zero-in on exactly what you want

**Save & Personalize**

• Tap “Save” on any business profile to build your own favorites list

• Chat with our AI-powered assistant to get recommendations tailored to your past saves, ratings, and reviews

**Book with Confidence**

• View each business’s opening hours and available hourly slots

• Book your spot—date and time—in just a few taps, with confirmation sent immediately

**Rate & Review**

• After your visit, rate and leave a short review to help fellow travelers

• See aggregate ratings and read others’ experiences before you go

**Seamless Onboarding & Support**

• Sign up in seconds with just your email and password

• Our step-by-step online guide walks you through any feature, and in-app tips pop up right where you need them

Under the hood, GO’s backend tracks your interactions, powers personalized suggestions via OpenAI, and keeps business profiles up-to-date with easy image uploads to S3. Whether you’re a traveler looking for your next adventure or a local business showcasing your services, GO delivers a frictionless, end-to-end booking experience—all you need is your smartphone and a sense of curiosity.

## **Product Functions**

Go is designed to feel less like an app and more like a personal travel companion. Everything we built focuses on helping users discover, plan, and truly experience Lebanon in a deeper, easier way. Below are the core features we have implemented so far, along with a few exciting “stretch” features we’re aiming to roll out in future updates.

**Major features**

1. **Home Screen with Destination Categories**

From the moment users open Go, they’re welcomed by a clean and friendly home screen that organizes destinations into easy-to-navigate categories. Whether someone is in the mood for a beach escape, a historical adventure, a hike through nature, or a cozy mountain village, they can find it quickly without getting lost in endless scrolling.

1. **AI Chatbot for Real-Time Assistance**

Planning a trip often comes with lots of small (but important) questions. Our built-in AI chatbot gives users a way to ask anything—best places to visit in winter, how to get to a certain village, tips for cultural etiquette—and get helpful answers immediately. It feels less like using an app and more like texting a local friend for advice.

1. **Interactive Map Page**

We knew we had to include a dynamic map that shows users exactly where destinations are, how far they are from each other, and what’s nearby. The map isn’t just a GPS tool—it’s a discovery tool. Users can tap, explore, and even build their own custom travel routes based on what sparks their interest.

1. **Personalized Recommendations and Save Functionality:**

Go learns with every tap. By tracking users’ search history, preferences, and saved destinations, the app suggests new places that match their interests—whether it’s boutique wineries, mountain trails, or hidden ruins. Users can also save favorite spots to their personal profiles for easy access when planning their trips.

**Stretch Features for Version 2.0**

1. **User Itinerary Planner**

One of the features we’re excited to introduce is a **smart itinerary planner**. Instead of just bookmarking destinations, users will be able to create full-day or multi-day travel plans inside the app. *Go* would automatically suggest the best route, optimal timing based on opening hours, and even nearby food stops—making trip planning as effortless as possible.

**2. Social Sharing and Travel Journals**

We’re also planning to add a feature where users can **create and share their own travel stories**, complete with photos, reviews, and personal tips. This not only helps future travelers but also turns *Go* into a community-driven platform where experiences are passed from one explorer to another.

## **User characteristics**

Whether they are foreign visitors or locals, Go is made to accommodate anyone who wants to learn more about Lebanon. Go’s development team has made the app extremely user-friendly by focusing on reducing user participation to streamline navigation and consumption. Therefore, Go Local customers don't need any specific expertise or abilities beyond using a smartphone.  
  
  
Ease of Use: • User-Friendly Interface: Go has an easy-to-use interface that leads users through a number of services, such as booking experiences and looking for nearby attractions.• Basic Requirements: To use Go efficiently, all you need is a basic understanding of smartphones, including how to use touch interfaces and GPS features for location-based services.

## **Non-functional Requirements**

Non-functional criteria are crucial for ensuring that the Go app fulfills performance and compliance standards while providing smooth user experience. These criteria cover product qualities, organizational alignments, and external limitations.

1. **Product Requirement:**

•Usability: The app's design is user-friendly, straightforward, and engaging, ensuring simplicity of use. Navigation via the app's parts is simple, increasing user engagement.

• Portability: Users may access their account from numerous devices, providing a seamless experience across platforms. Accounts can accommodate several user profiles, catering to varying interests and demands within the same family or group.

• Efficient: Quick app startup in under 5 seconds, reducing wait times. Stability is a top emphasis, with mechanisms in place to avoid unexpected crashes and ensure seamless functioning.

•Reliability: Reliable customer support is available to address feedback, and any issues users may encounter.

1. **Organizational Requirements:**

Standards:

•The app adheres to international standards and regulations concerning user safety and data privacy.

Delivery:

•Initial launch will be in Lebanon, followed by a global rollout, ensuring the app is completely functional at launch—no semi-finished products.

Implementation:

•The app is available for both iOS and Android devices to maximize accessibility

1. **External Requirements:**

Privacy:

•Location access will be requested only in future updates and with explicit user permission

•Passwords are obscured once set, enhancing security.

•The app will not access or save photos or utilize the device’s microphone or camera without user consent.

License/Copyright Issues:

•All content, including place descriptions and images, will respect copyright laws, ensuring all published material is legally compliant.

Social Issues:

•Designed to be accessible and user-friendly, the app accommodates users with limited tech skills or disabilities.

Ethical Issues:

•User data security is paramount, with encryption used for all sensitive data to protect user privacy

Legal Issues:

•Complies with local and international data protection laws, including provisions for user data deletion upon request.

## **Domain Requirements**

For the Go app to work properly, it must fulfill certain requirements designed for the Lebanon tourist industry:

• Language Support: To provide both domestic and foreign users, the app will offer text and content in Arabic and English.

• Geographic Categorization: For easier navigation, destinations will be grouped by kind (such as natural parks or cultural attractions)

. • Location Specificity: To facilitate trip planning, each venue or event will be associated with a particular geographic location.

• Consistent Listings: Unless absolutely required, the app will keep listings consistent across brands. •

Content Focus: All information will be solely about travel, leaving out irrelevant commercial services.

• Information Validity: To guarantee its correctness and applicability, site information will be updated on a regular basis.

• Information Validity: To guarantee its correctness and applicability, site information will be updated on a regular basis.   
  
These simplified specifications guarantee that Go effectively facilitates travel exploration in Lebanon while giving users reliable and helpful information.

## **Functional Requirements Specification**

To make *Go* a fully working, user-friendly travel companion, we identified a range of important features that the system must deliver. These functionalities are organized according to the user's journey and needs, covering both regular users, business accounts, and admin control.

1. **Authentication and Access Control**

* New users must be able to register by providing their email and password, creating their personal account.
* Users must be able to log into their accounts securely and receive an access token to stay connected.
* Administrators have a special login portal to access admin-only controls.
* Users should be able to refresh their session tokens easily to stay logged in, or log out safely when needed.
* Most features are protected — meaning users must be logged in to access them, ensuring secure interactions at all times.
* Special permissions must be checked for business accounts and admin activities, ensuring no one accesses areas they shouldn't.

1. **User Profile Management**

* Every user can view and update their personal profile—this includes editing basic information like their name and email.
* Business users can also update their business-specific profiles, such as adding a business description and services.
* Users should be able to upload their personal avatars, while businesses can upload a main cover image for their profile.

1. **Business Profiles and Destination Listings**

* Users should have access to a full list of destination categories, making it easy to browse by type (nature, history, adventure, etc.).
* All business profiles (destinations) are available to explore, with options to filter them by name, category, or region.
* A detailed page is available for each business, showing images, descriptions, ratings, and available booking slots.
* The system should also be able to group businesses by status (normal, business account, banned) for admin purposes.

1. **User Activities: Saving, Rating, Reviewing, and Booking**

* Logged-in users can save favorite destinations for future trips, and unsave them whenever they wish.
* Users should be able to rate destinations with a score and write reviews based on their experiences.
* The platform must track whether a user has already rated a destination to avoid duplicate entries.
* Users can book available slots for activities or stays directly through the app.
* A history of user bookings and saved destinations should be easily accessible within their profiles.

1. **Booking Management**

* Users can create new bookings by selecting a date and time slot for a service or destination.
* Businesses must be able to view all bookings made by customers for their services.
* Users should also have access to a clear list of all their own past and upcoming bookings.

1. **Image Handling**

* Users and businesses should be able to view and manage uploaded images attached to their profiles.
* The system should allow uploading new images (single or multiple uploads), supporting both regular pictures and specialized formats (like 3D images).
* There must be an option to delete existing images if users or businesses want to update their profiles.

1. **Admin Management and Monitoring**

* Admins need to **ban or unban users** based on activity, ensuring the platform stays safe and trustworthy.
* The admin dashboard should provide **insights and statistics**, like the total number of users, number of businesses, and number of banned accounts.
* Admins should have access to a **user table** where they can **filter, search, and take actions** directly, like banning or unbanning users.

1. **AI Chatbot and Smart Recommendations**

* The app should offer personalized destination recommendations based on the user’s past activity, helping users find places they'll love.
* The AI chatbot allows users to ask questions about travel, destinations, events, and more, providing real-time, helpful responses.
* The system should be able to detect names, places, and categories from user queries to improve the chatbot’s answers and recommendations.

1. **Other System Operations**

* There should be a system health check to make sure the application is running correctly at all times.
* Database seeders and migration tools are necessary to initialize important data like user roles, destination categories, and activity types when setting up the app.
* If users visit a wrong or broken link, they should be redirected to a friendly 404 page with a clear option to return to the home screen.

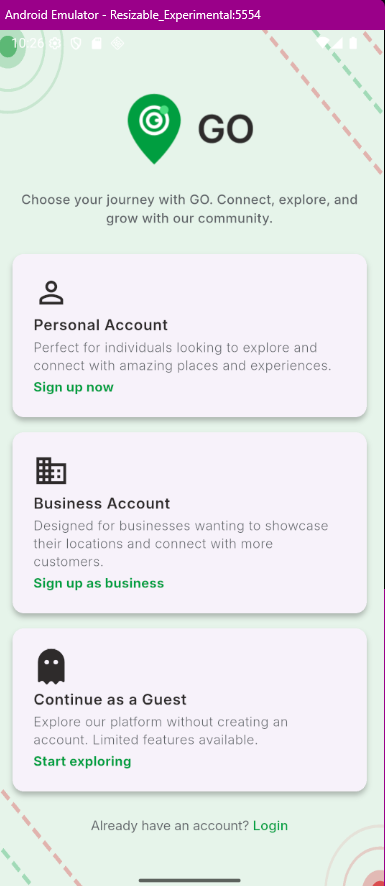
Each of these functional requirements ensures that *Go* delivers a complete, secure, and enjoyable experience for travelers, businesses, and administrators alike — making it not just another tourism app, but a trusted companion for anyone discovering the beauty of Lebanon.

# **Chapter 4. Project Design**

## **User Interface Prototype**

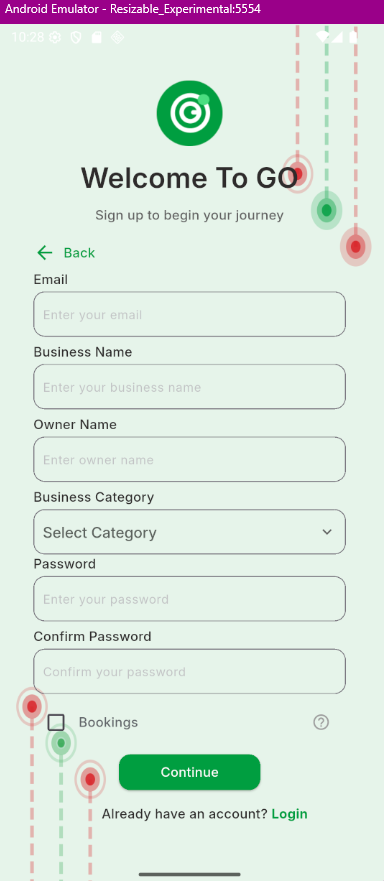
Designing a prototype is an essential phase in the development of our "Go" platform as it allows our development team to identify which features are operational and which require further refinement. Additionally, the user interface prototype is crucial for testing the workflow of "Go" across multiple scenarios. This step is particularly important to ensure that common use cases are streamlined for efficiency and ease of use. As highlighted in our discussions, and as demonstrated in our prototype, "Go" is designed with a highly intuitive user interface, enabling users to navigate the application effortlessly. The prototype effectively illustrates all the use cases of "Go," ensuring that each function is accessible and user-centric.

A screenshot of a login form

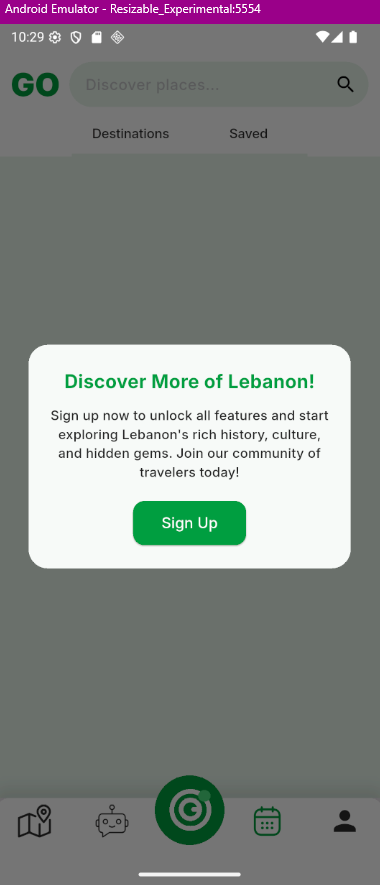
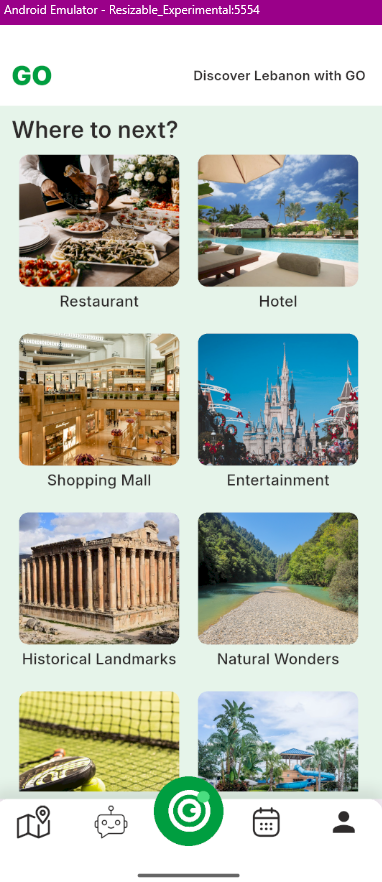
AI-generated content may be incorrect.

Sign up options Sign up page

A screenshot of a login form

AI-generated content may be incorrect.

Sign up for business Accounts Login Page



Pop Up for guests not being able to use Home pagee

Available features

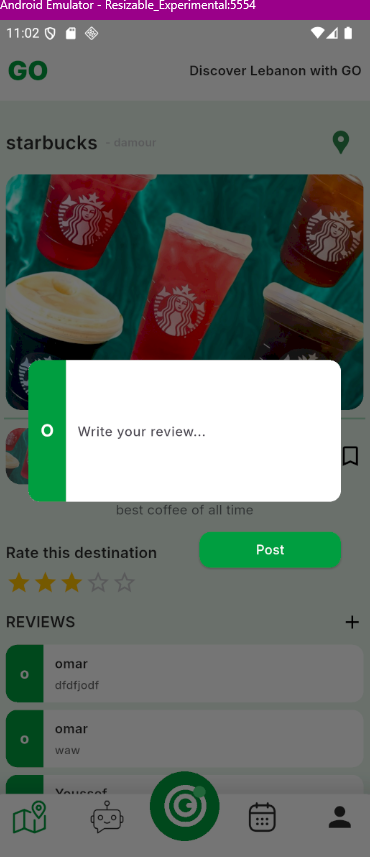
A screenshot of a phone

AI-generated content may be incorrect.A screenshot of a phone

AI-generated content may be incorrect.

Destinations page Single Destination Page

A screenshot of a calendar

AI-generated content may be incorrect.

Review Booking

A map with a location on it

AI-generated content may be incorrect.A screen shot of a phone

AI-generated content may be incorrect.

ChatBot Map

A screenshot of a phone

AI-generated content may be incorrect.



Map

Saved destination

A screenshot of a phone

AI-generated content may be incorrect.A screenshot of a login form

AI-generated content may be incorrect.

Profile Business Profile

A screenshot of a phone

AI-generated content may be incorrect.A screenshot of a phone

AI-generated content may be incorrect.

Monthly payment Business Profile modification

A screenshot of a phone

AI-generated content may be incorrect.A map with a red point on it

AI-generated content may be incorrect.

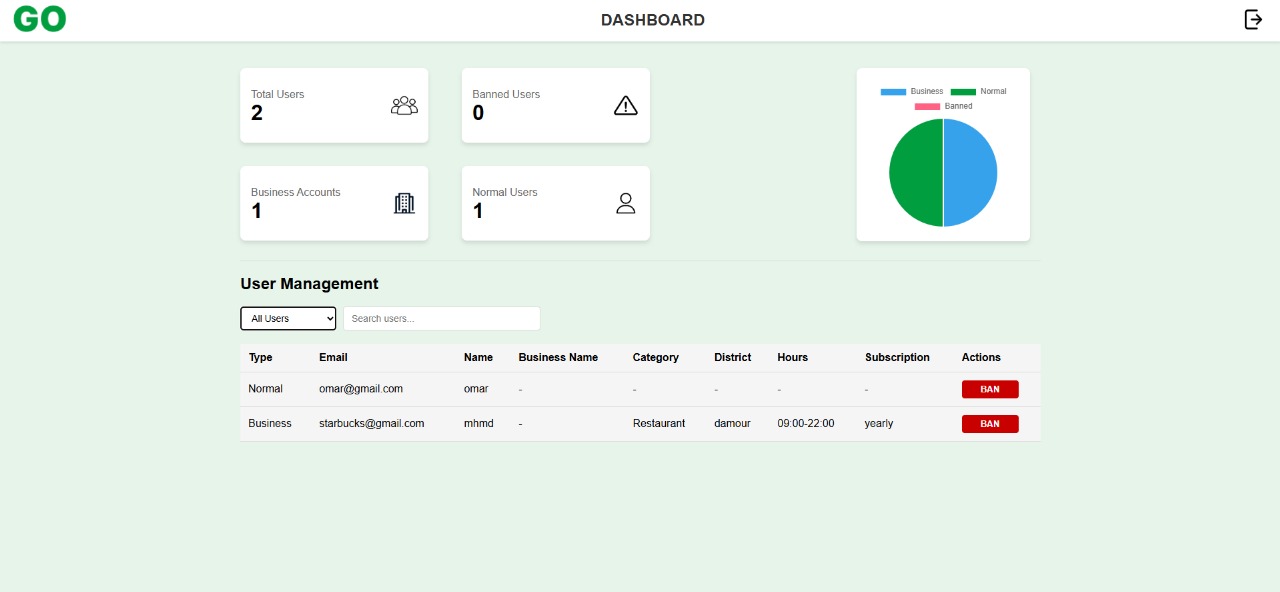
Editing business location Adding, Deleting and modifyion

pictures and bio

A map of the country

AI-generated content may be incorrect.

Dashboard login



Dashboard holding info about users

## **Data Flow Diagram**

DFD describes the processes that are involved in a system to transfer data from the customer and tour company to the system. In other words, a data flow diagram shows how data flows throughout a system. Go is an app that depends on data from multiple aspects. First and foremost, a user should create a touristic account. To do so, the user can use credentials or sign in through a third-party application. When the user signs in, all the data stored in our system of all the recommendations in his/her search, Wishlist, or Chatbot will be in display. We also need to store data regarding the user’s favorite items and shopping list. Therefore, when a user uses our app, multiple types of data flow in and out of the app’s database

**Level 0**

This diagram illustrates a system architecture for a platform involving three main entities: Customer, System, and Admin, along with an additional entity, the Tour Company. Here’s a detailed explanation:

---

1. **Customer:**

The Customer interacts with the System for the following purposes:

* Browse: Customers can explore the available destinations or services offered by the system.
* Book: Customers can make bookings for tours or services.
* Feedback: Customers provide reviews or feedback on their experiences.
* Wishlist: Customers can save destinations or services for future reference.
* Booking Confirmation: Customers receive confirmations for their bookings through the system.
* Reports: Customers may access their booking history or other personalized data.

1. **System**

The System acts as the central platform that connects all entities. It manages interactions between the Customer, Admin, and Tour Company, enabling:

* Handling bookings, feedback, and user data from customers.
* Sending booking confirmations and generating reports.
* Managing destinations, inventory, and overall platform data.
* Generating reports and analytics for other entities.

1. **Admin**

The Admin has full control of the System and performs administrative tasks:

* Manage Users: Admins handle customer and tour company accounts.
* Manage Bookings: Admins oversee booking-related issues or conflicts.
* Generate Reports: Admins can produce detailed reports on system performance, user data, or other metrics.

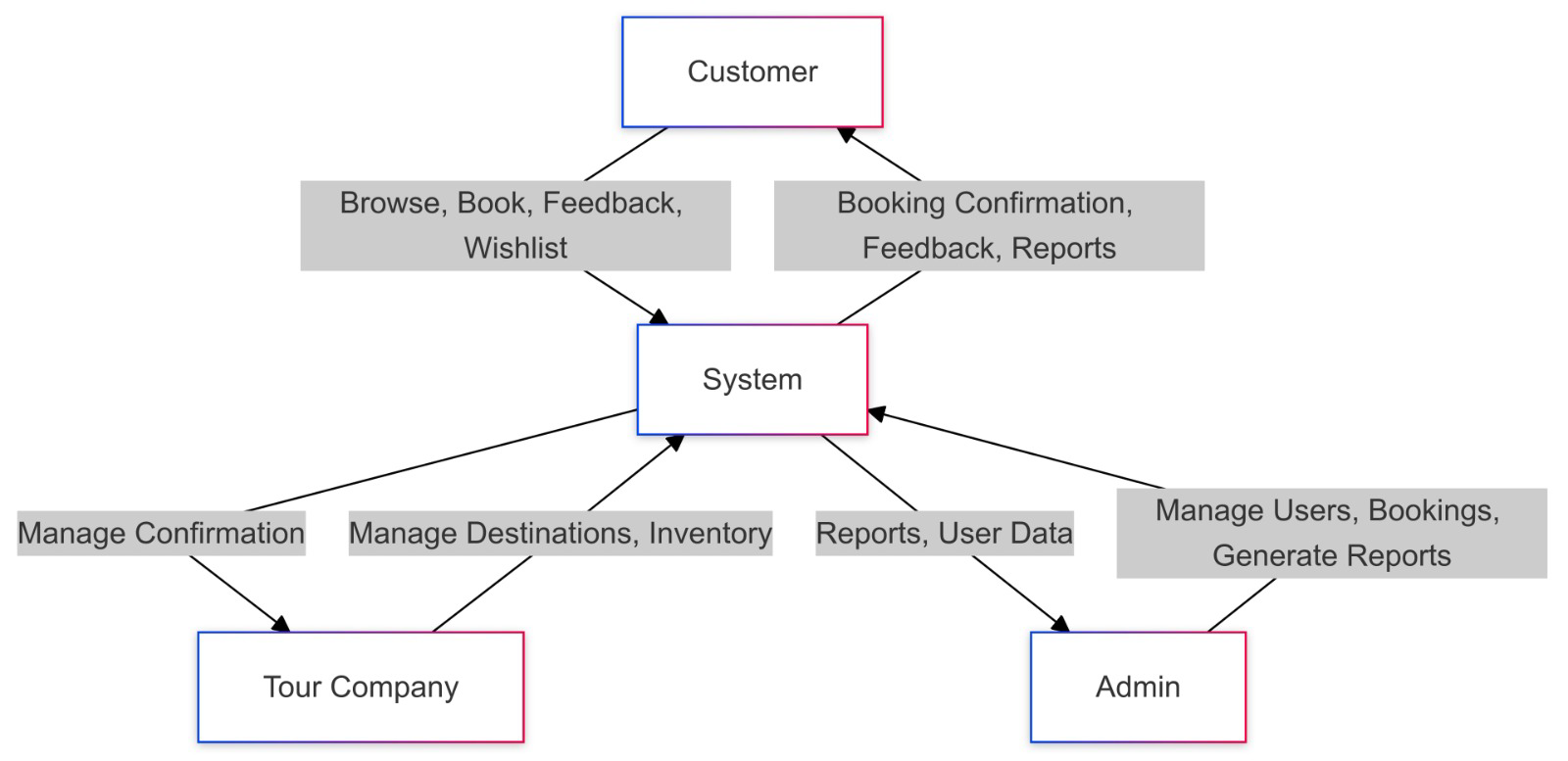
1. **Tour Company**

The Tour Company provides services and interacts with the System for the following:

* Manage Confirmation: Ensures that customer bookings are confirmed or processed.
* Manage Destinations & Inventory: Updates the system with available destinations, tours, and resources.
* Reports: Accesses system-generated analytics related to bookings and user feedback.

**Interactions**

* Customer → System: Customers interact with the system for browsing, booking, and feedback purposes.
* System → Tour Company: The system ensures the tour company is updated on bookings, destinations, and inventory.
* System → Admin: The admin accesses the system to manage users, bookings, and reports.
* System ↔️ Tour Company and Customer: The system acts as a mediator to streamline communication and data flow. Customer, System, and Admin, along with an additional entity, the Tour Company



**Level 1**

DFD level 1 appears to represent a system architecture or process flow for managing various components of a booking and management system, likely related to tourism, travel, or accommodations. Here's a breakdown:

Main Components:

1. **Customer:**

The primary user who interacts with the system.

Can submit booking requests, provide feedback, and interact with the wishlist.

1. System:

Responsible for handling user credentials, processes, and interactions with the database.

1. Admin:

Manages user data, reports, and destination details.

1. **Database:**

Acts as the central repository for all data related to bookings, feedback, wishlist, user credentials, and destinations.

**Key Processes:**

1. **User Login Process:**

Customers submit credentials for login.

The system validates credentials and confirms login.

1. **Booking Management Process:**

Customers submit booking requests.

The system validates bookings and stores the booking data in the database.

1. **Feedback Submission Process:**

Customers submit feedback.

Feedback is validated and stored in the database.

1. **Wishlist Management Process:**

Customers save destinations to a wishlist.

Wishlist data is validated and stored in the database.

1. **Destination Management Process:**

Admin or tour companies manage destination information.

Destination data is stored in the database.

1. **Admin User Management Process:**

Admin retrieves, displays, and manages user data.

Can also fetch resort/destination data as needed.

1. **Report Generation Process:**

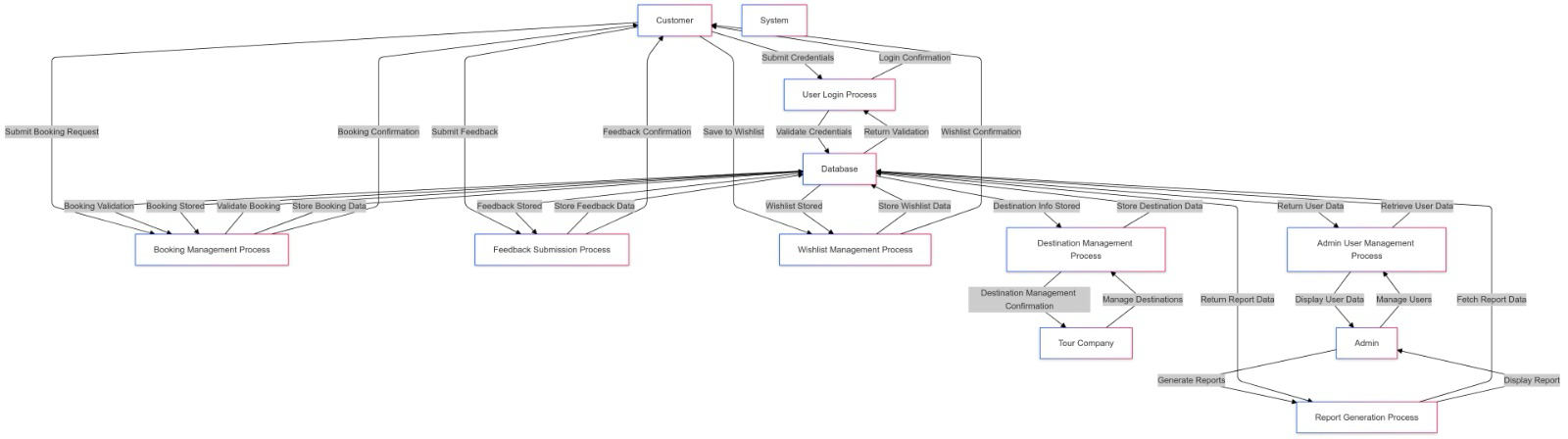
Admin generates reports based on stored data.

Reports are displayed for analysis or decision-making.

**Flow of Actions:**

Customer interacts with the system by submitting requests (e.g., bookings, feedback).

The System validates and processes requests, storing the results in the Database.

The Admin oversees the data, managing destinations, users, and generating reports for organizational needs.

## **Use Case Diagram**

A use case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved. Our proposed system consists of package of services, this package is going to be explained in the upcoming figure.

Use Case Diagram Explanation

Actors:

Customer: Interacts with the system for functionalities like searching destinations, submitting feedback, booking modifications, and chatbot requests.

Tour Company: Manages destinations, inventory, and places in the system.

Admin: Oversees system operations, generates reports, and manages bookings and users.

Use Cases:

Use cases represent specific actions the actors can perform, like Submit Feedback, Manage Tours, or Generate Reports.

Relationships such as <<include>> and <<extend>> denote dependencies and optional functionalities. For example:

User Signup includes Fill User Form because filling the form is a necessary part of signing up.

Browse Destination and Interest extends Add Booking, as interest tracking might lead to a booking.

A diagram of a chat

AI-generated content may be incorrect.

A diagram of a chat

AI-generated content may be incorrect.

A diagram of a chat

AI-generated content may be incorrect.

## **Database Diagram**

**Tables Overview:**

The user login table is the central table that stores login credentials for all roles (customer, tour company, admin). It includes a role field to differentiate between

types of users.

The customer, tour company, and admin tables link to user\_login via a user\_login\_id foreign key, ensuring a one-to-one relationship with their login credentials.

The booking table tracks booking details, including relationships to customer and destination.

The destination table holds data about locations, with references to the tour\_company responsible for managing them.

Other tables like feedback, wishlist, inventory, place, report, chatbot\_request, and user\_destination\_interest handle specific functionalities like user feedback, saved destinations, and interest tracking.

Relationships:

Customers interact with destinations via bookings, feedback, and wishlists.

Tour companies manage destinations, places, and inventory.

Admins oversee reports, users, and bookings.

This database ensures a normalized structure, avoids redundancy, and supports scalability for a mobile application.



## **UML Diagram**

**A screenshot of a computer

AI-generated content may be incorrect.**

The classes required for the class diagram are the same as the collections assigned with their fields as it is in the NOSQL scheme; these are Customer, Destination, Tour Company, Booking and much more classes. These said classes have the same attributes and methods as above. For example, the class Destination will have id, name, description, and interest. In addition, the relationships that exist between these classes are the same as in the above scheme.

Customer, TourCompany, and Admin: Represent the roles in the system, each with their attributes (name, email, password) and behaviors (methods like login, signup, and manage-specific functionalities).

Booking: Tracks bookings between customers and destinations.

Destination: Contains attributes for the destination's details and a method for updating interest metrics.

## **Sequence Diagrams**

This diagram demonstrates the flow of communication between the Customer, System, Database, Tour Company, and Admin to facilitate core functionalities like booking, feedback submission, Wishlist management, and admin control.

This is a sequence diagram that represents interactions between different components or actors in a system. Here is a breakdown of its elements:

**Actors**

1. Customer: The end user interacting with the system.

2. System: The software platform handling user requests.

3. Database: The backend storing data like user credentials, feedback, bookings, etc.

4. TourCompany: Manages tour-related data (minimal role here).

5. Admin: Manages administrative tasks like user and report management.

**Interactions**

1. **Login Process**

Customer provides email and password.

System validates credentials with the Database.

Login status (success/failure) is returned to the Customer.

1. **Submit Feedback**

Customer submits text and rating.

Feedback is stored in the Database.

Confirmation is sent back to the Customer.

1. **Browse and Book Destinations**

Customer browses available destinations.

System retrieves a list from the Database.

After selecting, booking details are validated and stored.

1. **Modify Booking**

Customer modifies booking (e.g., changing the date).

Existing booking details are retrieved and updated in the Database.

Modification is confirmed.

1. **Save to Wishlist**

Customer saves a destination to their Wishlist.

Wishlist data is stored in the Database.

Confirmation is provided.

1. **Submit Rating and Review**

Customer submits a review and rating.

The review is stored in the Database, followed by a confirmation.

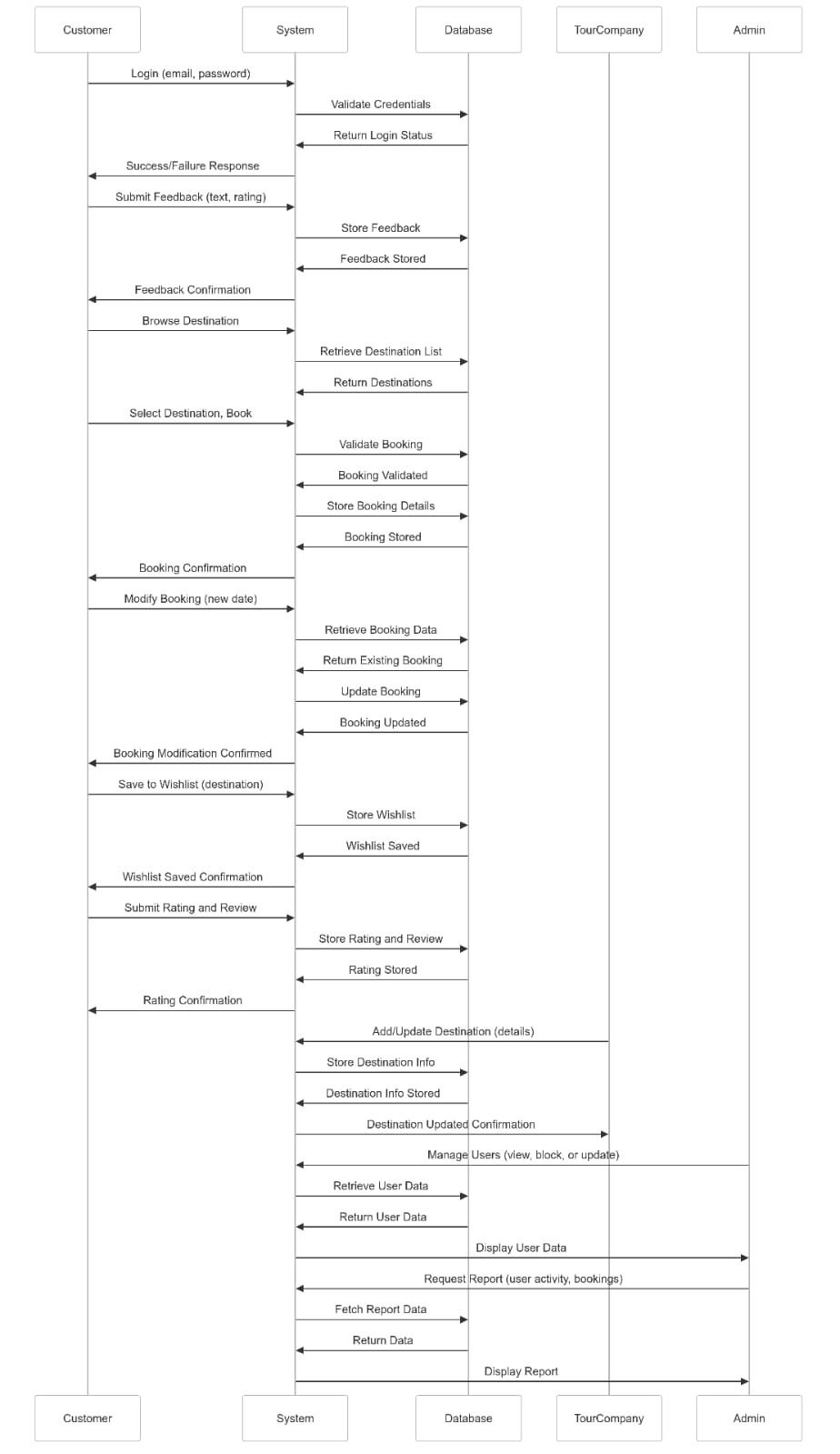
1. **Admin Actions**

**Admin can:**

Add or update destinations (stored in the Database).

Manage users (view, block, or update user data).

Request reports (e.g., user activity, bookings) from the Database.



# **Chapter 5. Implementation**

## **Software Architecture**

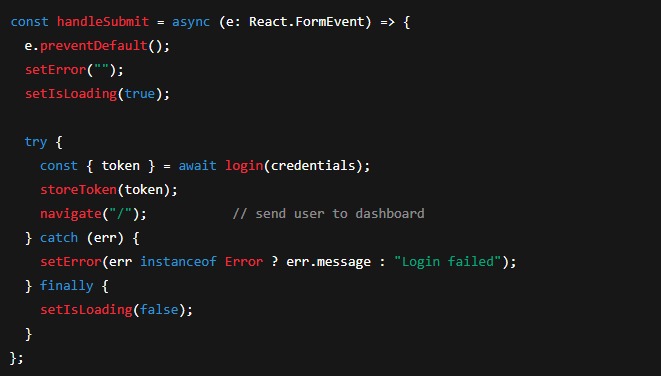
## **Technology Stack**

programming languages, frameworks, databases, front-end and back-end tools, and APIs

Explain why you chose these tools and languages, as well as why they are suitable for use on this project.

## **Front-End Implementation**

LoginForm.tsx – Handling form submission



Prevents the default form post and shows a loading state.

Calls our auth API (login) with the user’s email/password.

On success, stores the JWT in localStorage and redirects to /.

On failure, displays the error message.

requestApi (utils/requests.ts) – Centralized API client

A screen shot of a computer screen

AI-generated content may be incorrect.

All HTTP calls go through this single function—so token injection, error logging, and content-type handling are consistent across the app.

ProtectedRoute.tsx – Guarding private pages

A computer screen with white text

AI-generated content may be incorrect.

Wraps any routes that require login. If no valid token is found, redirects back to the login page.

DashboardStats.tsx – Displaying summary cards

A computer screen shot of a computer code

AI-generated content may be incorrect.

Renders four consistent “cards” showing key metrics, driven entirely by the stats prop passed from the Dashboard page.

how the Dashboard page ties front-end to back-end:

A computer screen with text on it

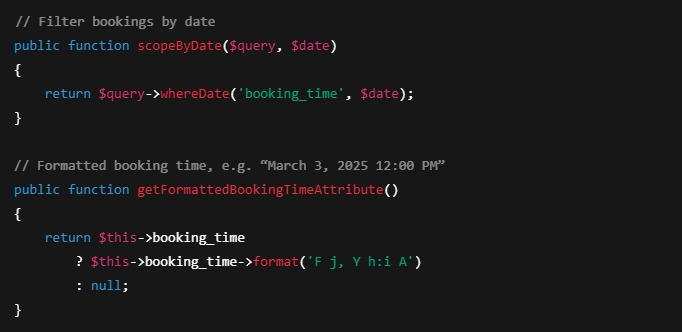
AI-generated content may be incorrect.

fetchUsers() calls our Laravel endpoint /api/destinations/grouped.

The result populates both the stats cards and the table of users.

## **Back-End Implementation**

Booking.php (Model) – Scopes & Accessors



Scopes let controllers write Booking::byDate($d)->get().

Accessors automatically add a formatted\_booking\_time field when the model is serialized.

BusinessProfile.php – Generating hourly slots

A computer screen with text on it

AI-generated content may be incorrect.

Computes on-the-fly an array of “09:00”, “10:00”, … until closing—so the front-end can render exactly the available booking times.

**ActivityController.php – Saving a “save” action**

A computer screen shot of a program code

AI-generated content may be incorrect.

Validates the request, then writes a UserActivity record of type “save”.

Uses our ApiResponseService for consistent JSON responses.

**OpenAIService.php – Generating recommendations**

A computer screen shot of a program code

AI-generated content may be incorrect.

Pushes a summary of your past saves/ratings to OpenAI and gets back a simple bullet-list of your favorite categories—no in-house ML expertise required.

**External APIs**

**OpenAI Chat Completions**

Used to generate personalized destination recommendations, power the in-app chatbot, and extract entities (destination names, categories, districts) from user queries. Accessed via Guzzle HTTP client in OpenAIService.

**AWS S3 REST API**

Handles storage and deletion of user and business images (profile pictures, gallery uploads). Invoked through Laravel jobs (UploadImageToS3, DeleteImageFromS3) using the AWS SDK for PHP.

**Internal REST Endpoints**

All routes are prefixed with /api and most are protected by JWT (auth:api), with additional middleware for business or admin actions where noted.

**Authentication & Profile**

POST /api/register — create a new user

POST /api/login — obtain a JWT for normal users

POST /api/admin/login — obtain a JWT for admin users

POST /api/logout & POST /api/refresh — invalidate or refresh the token

GET /api/me — fetch the logged-in user’s profile

PUT /api/profile/edit — update personal profile fields

**Image Management**

POST /api/profile/upload-profile-image — upload a user avatar

POST /api/profile/upload-business-main-image — upload the main business cover image

GET /api/users/{user}/get/images — list all images for a user

POST /api/users/{user}/images — batch-upload one or more images

DELETE /api/users/{user}/delete/images — delete specified images

**Business Profiles & Discovery**

GET /api/categories — retrieve all destination categories

GET /api/destinations — list all business profiles

GET /api/destinations/grouped — get counts and lists of active normal, active business, and banned users

GET /api/destinations/name/{name} — find a business by name

GET /api/destinations/{userId} — view a single business’s profile, including slots, images, reviews, and rating

GET /api/destinations/category/{category} and /district/{district} — filter businesses by category or district

GET /api/destinations/bookings/{businessUserId} — show all bookings for a particular business

GET /api/destinations/reviews/{businessUserId} and /rating/{businessUserId} — view that business’s reviews and average rating

**User Activities**

POST /api/activity/save, /unsave, /rate, /review, /book — record a save, remove a save, submit a rating, post a review, or book a slot

GET /api/user/check-rated/{businessUserId} — check whether the current user has already rated that business

GET /api/user/{userId}/bookings — list all bookings made by a given user

GET /api/user/{userId}/saved — list all destinations that user has saved

**Admin Controls**

POST /api/users/ban and /api/users/unban — ban or unban a user (requires AdminCheck middleware)

**AI & Recommendations**

GET /api/recommend-destinations/{userId} — run the user’s activity history through OpenAI to get top-category suggestions

POST /api/{userId}/chatbot — send a free-form query to the travel assistant and return its reply

**PHP (Composer) Dependencies**

laravel/framework — the core MVC framework that provides routing, Eloquent ORM, queues, jobs, and Artisan CLI tools.

aws/aws-sdk-php — AWS SDK for PHP, used by Jobs to upload and delete images in S3.

guzzlehttp/guzzle — HTTP client for calling the OpenAI API and other external services.

tymon/jwt-auth — integrates JWT-based authentication into Laravel’s API guard.

**JavaScript (npm/Yarn) Dependencies**

react & react-dom — build the interactive single-page application.

react-router-dom — client-side routing for login, dashboard, and error pages.

axios — make promise-based HTTP requests to the Laravel back end.

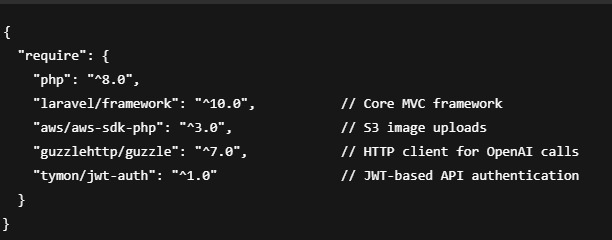
typescript — static typing for more reliable, self-documenting front-end code.

vite — lightning-fast development server and build tool for the React app.

chart.js & react-chartjs-2 — render the dashboard’s pie chart of user distribution.

These APIs and packages work together to form GO’s complete stack: a secure, scalable Laravel back end talking to a responsive, type-safe React front end—augmented by AI and cloud storage for a smooth, modern travel-booking experience.

**npm (JavaScript) Packages**



laravel/framework: scaffolds routing, models, controllers, migrations, queues.

aws/aws-sdk-php: abstracts S3 operations into dispatchable jobs.

guzzlehttp/guzzle: reliable HTTP for external API integration (OpenAI).

tymon/jwt-auth: secures API with stateless JSON Web Tokens.

**Composer (PHP) Packages**

A computer screen with white text

AI-generated content may be incorrect.

react & react-dom: build your single-page UI.

react-router-dom: manage URL-based navigation.

axios: promise-based HTTP client.

typescript: catches bugs at compile time, improves IDE support.

vite: hot module replacement & super-fast builds.

chart.js & react-chartjs-2: visualize data (e.g. the dashboard pie chart).

@mui/material & icons: polished, accessible components for the 404 error screen.

## **Deployment and Delivery**

Thought for a couple of seconds

Below is the documentation we’ll ship alongside the GO application. It walks through every screen, explains its purpose, and highlights the key features each page offers.

**1) Splash / Launch Screen**

As soon as you open GO, you see our branded splash screen. Its goal is to give a polished, welcoming first impression, reassure you that the app is loading, and reinforce our travel-focused identity before you land on the login page.

**2) Sign In / Sign Up Pages**

These screens get you into GO quickly, whether you’re a returning traveler or a new user.

* **Sign Up**: Enter your name, email, and a password of your choice. Behind the scenes we create your user record, assign the "Normal" role, and kick off any welcome workflows.
* **Sign In**: Enter your credentials to retrieve a JWT. Forgot your password? Tap “Reset” to receive a reset link by email.  
  Both forms validate input in real time and display inline errors—no guessing if you mistype your email or password.

**3) Discover / Home Page**

Once authenticated, you arrive at the Discover screen, which is GO’s main catalog of business profiles: restaurants, hotels, attractions, and more.

* **Category Tabs** let you switch between “Restaurant,” “Hotel,” “Entertainment,” etc.
* **District Filter** narrows results to your chosen neighborhood.
* **Search Bar** instantly filters by name as you type.  
  Everything is powered by calls to /api/destinations, so the list is always up to date.

**4) Destination Detail Page**

Tap any listing to see its full profile:

* **Photos Gallery** (uploaded to S3)
* **Business Name, Category & District**
* **Description** and **Average Rating**
* **“Save” Button** that writes a UserActivity record via POST /api/activity/save
* **Reviews & Ratings** fetched from /api/destinations/reviews/{businessUserId}

This page bundles everything you need to decide whether to book.

**5) Booking Page**

From the detail view, hit “Book a Slot” to open the booking screen:

* **Hourly Slots** (computed server-side by getAvailableBookingSlotsAttribute)
* **Date Picker** for selecting your visit date
* **Confirm Booking** button that sends a POST /api/activity/book request  
  Immediately after, you see a confirmation and your new booking appears in both your profile and the admin dashboard.

**6) Chatbot Page**

Tap the chat icon to ask GO’s AI assistant—powered by our OpenAIService.

* You type a free-form question (“Recommend a romantic dinner spot in Midtown”)
* Front-end calls POST /api/{userId}/chatbot, which relays your query to GPT-3.5-turbo
* You see the AI’s reply in real time, formatted as a simple list of suggestions

This keeps you in the app, never bouncing to a separate website.

**7) Favorites / Saved Page**

All the places you’ve tapped “Save” on show up here.

* **List View** with business name, thumbnail, and a “Remove” button
* Backed by GET /api/user/{userId}/saved and POST /api/activity/unsave  
  This makes it easy to revisit your shortlist when you’re ready to book.

**8) Profile & Settings Page**

Under your user avatar you’ll find:

* **Profile Info** (name, email, profile image upload via /api/profile/upload-profile-image)
* **Change Password** form
* **Privacy & Notifications** toggles for email alerts
* **Logout** button that clears your token and returns you to Sign In

Everything here issues clean JSON requests to our API and gives immediate feedback.

**9) Admin Dashboard (for Business Owners & Admins)**

If you log in as a business or admin user, you land on a special dashboard:

* **Stats Cards** show total users, active businesses, and banned accounts (via /api/destinations/grouped)
* **User Distribution Chart** (Pie chart using Chart.js)
* **User Management Table** where you can filter by role or search by name/email
* **Ban / Unban** buttons that call POST /api/users/ban or /api/users/unban

This page is built with React components DashboardStats, UserDistribution, and UserTable, all fed by our /api/destinations/grouped endpoint.

**10) Help & About Page**

Accessible from the main menu. It contains:

* **Terms & Conditions** (static content pulled from our web server or embedded Blade view)
* **User Guide** with links to each feature overview
* **Contact & Support** links that send you to our online support portal
* **Version Info** showing the current React and Laravel versions

This ensures every user can find help without hunting through the app.

**Delivery Format**  
We’ll package this as a beautifully formatted PDF (or web-hosted HTML) with screenshots of each page, code references, and descriptions of every button, API call, and UI flow—so anyone (developer, tester, or end user) can understand GO’s complete feature set at a glance.

## **CS Knowledge Application and Learning**

* **Formal Computer Science Theory:** While GO doesn’t directly model Turing machines or build custom automata, the underlying frameworks (React’s virtual DOM diffing, Laravel’s routing middleware) implicitly leverage state machines and event-driven automata concepts. For example, React’s reconciliation algorithm can be seen as a tree-transformation procedure, akin to formal tree transducers.
* **Advanced Data Structures:**
  + **Hash Tables:** Eloquent’s in-memory collections use PHP’s associative arrays under the hood—hash tables that provide average-O(1) lookup for models cached in memory.
  + **Trees:** Both the React component tree and Laravel’s service container bindings rely on hierarchical structures. React diffing walks the component tree to apply minimal updates, and Laravel resolves nested dependencies via a tree of service providers.
* **Algorithms:**
  + **Search & Filter:** Front-end filtering of user lists uses simple linear scans over arrays, yielding O(n) complexity—which remains performant for typical user volumes (<10,000 entries).
  + **Sorting:** JavaScript’s Array.prototype.sort and SQL’s ORDER BY handle ordering needs.
  + **Graph Algorithms:** We don’t implement path-finding or complex graph traversal, but Laravel’s Eloquent will generate optimized SQL joins when querying related models, leveraging B-tree indexes in PostgreSQL.
* **Computer Security:**
  + **Authentication:** We use JWT (via Tymon’s package) for stateless, signed tokens—protecting against session hijacking and CSRF in API calls.
  + **Authorization:** Role-based middleware (AdminCheck, CheckBusinessAuthorization) enforces least-privilege access.
  + **Input Validation & Sanitization:** Every controller action applies Laravel’s validation rules to prevent SQL injection, XSS, and other injection attacks.
  + **HTTPS Enforcement:** In production, VOIP environments are configured to require TLS for all front-end → back-end traffic.
* **Database Systems:**
  + **Conceptual & Relational Schema:** Our database schema—defined via Laravel migrations—reflects a normalized, relational design: Users, Roles, Categories, BusinessProfiles, Bookings, Activities. Foreign keys enforce referential integrity.
  + **Queries:** We rely on Eloquent scopes (scopeByUser, scopeActive, etc.) and raw SQL indexes on foreign keys for O(log n) lookups at scale. Aggregate queries (counts, averages) use SQL’s built-in aggregation functions for efficiency.
* **Other Courses:** Concepts from web engineering and software design patterns (Model–View–Controller, repository patterns) are directly applied. Event-driven programming from the operating systems course informs our use of queues and asynchronous jobs.

**CS Emerging Technologies**

* **AI Integration:** GO leverages OpenAI’s GPT-3.5-turbo to power destination recommendations and the interactive chatbot. This offloads NLP and recommendation modeling to a managed API—demonstrating practical adoption of machine learning without in-house model training.
* **Machine Learning:** While we don’t train models locally, our architecture is extensible: activity logs could feed into custom collaborative-filtering pipelines in future releases.
* **Image Processing:** We store and serve user-uploaded images via AWS S3; thumbnails and optimization could be added later with AWS Lambda image-processing triggers.
* **Virtual/Augmented Reality:** Not in this phase, but the API-first design means we could later surface GO’s catalog in AR tour-guide apps without major backend changes.

**Self-Learning & Growth**

* **Workshops & Seminars:** Team members attended Laravel LiveWire workshops and React conferences to deepen expertise in component-based architectures and full-stack integration.
* **Labs & Tutorials:** We followed official Laravel and React tutorial series, as well as hands-on labs with Docker-ized local environments, to master containerized development and CI/CD pipelines.
* **New Technologies:** We prototyped Vite’s HMR (hot module replacement) to dramatically speed up front-end iteration, and evaluated AWS S3 vs. DigitalOcean Spaces for image storage costs and performance.

**Collaboration**

* We integrated GitHub Actions for automated testing on every pull request, ensuring code quality across front-end and back-end teams.
* Regular sprint planning and retrospectives kept everyone aligned: UI/UX designers, front-end engineers, backend developers, and QA analysts all contributed to backlog grooming and feature demos.
* We engaged with an external UX consultant to run user interviews on our wireframes—incorporating real-world travel-planning workflows into the final design.

Together, these CS foundations, emerging-tech applications, continuous learning, and cross-functional collaboration have powered GO from concept to a polished, production-ready travel assistant.

# **Chapter 6. Testing and Maintenance**

## **Testing**

Testing is a critical phase in the development of any software, ensuring that the application functions correctly and provides a positive user experience. Testing was conducted both during the development cycle and after the final implementation of the app. Several key factors were considered when testing the GO app:

Devices:

Testing was carried out on a variety of physical Android and iOS devices to ensure cross-platform compatibility. In addition, emulators were used for devices that we did not own, extending our coverage. This combination of physical and emulation testing helps achieve reliability while also expanding the testing scope. Special attention was paid to low-end devices, as they are commonly used by our target audience, including children, elderly users, and people with limited access to higher-end devices.

Unit Testing:

The primary objective of unit testing was to ensure that individual parts of the app work as expected in isolation. This makes it easier to identify and fix bugs in specific areas. Each unit was tested with a set of predefined inputs, including edge cases, to ensure that the app is robust and foolproof. Test cases were created for each feature, specifying the test case ID, description, priority (low, medium, high), prerequisites, execution steps, expected and actual outputs, test results, and post-conditions.

Example: A test case might test the accuracy of a place search feature in the app, where the expected output would be the correct location suggestions based on user input.

Integration Testing:

Once the individual units were verified, integration testing was conducted to ensure that all parts of the app work well together. This phase focused on ensuring that the interaction between the user interface, the backend system, and any external APIs (such as location services or event listings) functioned seamlessly. For example, we tested if the user’s location input was correctly communicated to the app’s search feature and if the results were accurately displayed from the database.

Usability Testing:

Usability testing involved sessions with a group of potential users to observe and analyze how they interacted with the app. This process helped identify any areas of confusion or difficulty that users might encounter. We asked participants to complete specific tasks to ensure the app was user-friendly. Based on their feedback, we made several design improvements to make the app more intuitive and accessible. Users' suggestions were incorporated into the final design to ensure a smooth and enjoyable experience.

In conclusion, the tests implemented in the development of the GO app ensure that it meets the necessary software requirements, including functionality, usability, cross-platform compatibility, and ease of use.

## **Maintenance**

The release of the GO app is just the beginning. Ongoing maintenance is crucial for ensuring that the app continues to provide a seamless user experience. In addition to fixing any issues that arise after the launch, regular updates will be implemented to enhance the app’s functionality and keep it compatible with the latest devices. To manage the development process and avoid confusion, we will utilize Git for version control. This allows team members to work simultaneously, track changes, and ensure everyone stays up-to-date.

As technology continues to evolve, the GO app will be frequently updated to remain compatible with new Android and iOS versions, devices, and user expectations. Regular bug fixes, feature enhancements, and performance optimizations will be part of the ongoing maintenance to ensure that the app remains competitive and reliable for users.

# **Chapter 7. Conclusion and Future Work**

With all the actions and choices made to create an application which we are pleased with, and which will fulfill its goal of enhancing users' travel experiences, creating Go has been a challenging but satisfying process. Go simplifies the process of discovering and exploring Lebanon's varied natural and cultural landscapes, allowing users to get ready for travel more quickly and engage with the local environment on a much deeper level. To help tourists make the most of their trips and avoid the very common disappointments that occur from missing important experiences due to inadequate planning or a lack of knowledge, Go provides regular alerts and tailored advice.

It has taken a lot of planning and execution to bring Go to life, including risk assessment, establishing good organizational procedures, ongoing internal and advisor engagement, and thorough testing of the app to ensure it is ready for release. Our dedication to upholding high standards and gradually enhancing functionality is unwavering.   
The application's initial release is just the start of what we want to see from it going forward. Future ideas include enabling customized itineraries that consider each person's interests and needs, as well as adding multi-user functionality to enable families or groups traveling together. Future updates will further enhance user convenience by incorporating real-time booking and availability checks for nearby events and attractions.

To meet the changing demands of its users and stay up to date with the most recent developments in technology and travel, our staff is dedicated to constantly improving Go. To improve everyone's travel experience in Lebanon and make it more accessible, pleasurable, and educational, we are constantly changing and adding new features.

# **Chapter 8. References**

<https://www.lonelyplanet.com/>

<https://kazderni.com/>

<https://www.figma.com/>

<http://draw.io>

<https://online.visual-paradigm.com/>

<http://studies.gov.lb/getattachment/Sectors/Tourism-and-Culture/2004/TOU-c04-3/TOU-C04-3.pdf>

<https://visit-lebanon.org/>

<https://maps.google.com/>

# **Appendices**

# 

Imagine planning a trip through Lebanon and deciding where to visit first. According to our chart, **Beirut** easily steals the show—about **40%** of travelers head straight for the capital’s buzzing streets, seaside cafes, and lively nightlife. But the heart of Lebanon isn’t all in one city: roughly a quarter of visitors each explore the sun-drenched vineyards and ancient Roman ruins of the **Bekaa Valley** or wander through **Tripoli**’s winding souks and centuries-old citadel. And if you’re looking for a quieter escape, the mountains around **Debbieh** still draw about **10%** of travelers, eager to swap the crowds for fresh air and lush pine forests. All told, Beirut may be the top pick, but Lebanon’s real charm lies in its wonderful variety—from vineyards to medieval lanes to mountain retreats.