PERSONALISED TRAVEL PLANNER AND RECOMMENDATION ASSISTANT

LI MINGJIE
ZAIHOSNITA HOOD

Faculty of Information Science & Technology,The National University of Malaysia,43600 UKM Bangi,Selangor,Malaysia

ABSTRACT

TravelBuddy is a web-based travel planning and recommendation system designed to help users manage their trips more efficiently. The system allows users to discover destinations, plan personalized itineraries, view real-time weather updates, and save favorite places. It aims to reduce the complexity of travel preparation by offering a centralized and user-friendly platform.

The system was developed using Vue.js for the frontend and Supabase for backend services, including authentication and cloud storage. Agile methodology was used to ensure iterative development and user-centered improvements. Key features include itinerary creation, destination recommendations, user profiles, comments, and admin management tools. TravelBuddy enhances travel planning by making the process simpler, smarter, and more organized.

1. INTRODUCTION

In recent years, the increased availability of digital technology and internet access has transformed the way people plan and organise their travel experiences (Buhalis & Law, 2008). Traditional travel planning methods, such as reading paper brochures or using travel agencies, are increasingly being replaced by online platforms offering real-time information and booking capabilities (Xiang, Wang, O' Leary, & Fesenmaier, 2015). Although websites such as TripAdvisor, Google Travel and Expedia offer helpful services, they often lack integration, each handling just one aspect of the travel process. Consequently, travellers often have to switch between multiple tools to coordinate transport, accommodation and activities, which results in inefficiencies and disjointed experiences (Neuhofer, Buhalis, & Ladkin, 2015).

Furthermore, the growing demand for personalised travel highlights the need for intelligent, centralised systems that can bring together the various elements of trip planning. This is particularly true for independent travellers, students and recreational users, who often lack the expertise or time to organise plans from a variety of sources. Without such a system, travel planning can be stressful, time-consuming, and prone to error (Gretzel, Sigala, Xiang, & Koo, 2015).

TravelBuddy was created in response to this problem as a comprehensive, smart trip-planning and recommendation system. The platform is designed to help users discover destinations, receive personalised trip recommendations, plan itineraries and view real-time weather conditions. Combining ease of use with robust functionality, the system enables travellers to make better and more informed decisions.

The project integrates modern technologies such as Vue.js for the front end and Supabase for the back end, employing a modular architecture and agile development methodology (Sommerville, 2016).

2. Problem Statement

Despite the existence of numerous digital platforms designed for travel-related activities, such as booking accommodation and arranging transport, as well as reading destination reviews, a significant gap remains in the provision of a fully integrated, customised travel planning experience. Most available platforms focus on specific functions and do not provide a centralised, end-to-end travel planning and management solution.

Initial user analysis and system benchmarking have revealed a number of important issues:

1. Lack of centralised itinerary management:

Users tend to manually track travel arrangements using general-purpose tools such as spreadsheets, note-taking applications or calendars. This ad hoc approach results in scattered information, editing issues, and no readily available history for future reference.

2. No integration of real-time contextual data:

Trip planning applications typically overlook the integration of real-time contextual data, such as weather predictions. Users are forced to switch to external websites to check the weather, which halts the planning process and inhibits proactive decision-making.

3. Lack of personalisation and data privacy:

Most free systems do not use user authentication or encrypted data storage. Consequently, users cannot retrieve or synchronise their itineraries between sessions or devices, and their personal information is either vulnerable or transient.

4. Inefficient Destination Discovery Process:

Many platforms list destinations in overwhelming formats without visual structure or recommendation logic. Users often struggle to explore new locations intuitively, especially without dynamic filtering or visual cues such as images and summaries.

These problems cumulatively create an ineffective, frustrating, and impersonal travel planning process—especially for independent travelers, students, and casual users who desire an easier and more intuitive solution.

3. OBJECTIVES

The following are the aims of creating the TravelBuddy system:

i. To create an online platform for custom travel planning and recommendations.

ii. To offer users itinerary management features, place discovery, and weather information.

iii. To improve the travel experience of users by incorporating real-time information and user feedback.

iv. To facilitate admin-level management to handle users, attractions, and content within the system.

4. METHODOLOGY

The methodology used in the development of this project is the Agile model. Agile was selected because it is iterative in nature, with ongoing user engagement and capacity to address evolving requirements during the lifecycle of the system. This approach focuses on incremental development, regular testing, and incorporation of feedback for ensuring delivery of a functional, user-oriented product (Gaurav & Pradeep, 2012). The Agile model allows flexibility with frequent improvements and quick adaptation to feedback without the time lag experienced in conventional models.

Agile Development Cycle

As illustrated in Figure 4.1, the Agile development cycle involves six main phases: Planning, designing, building, testing, releasing and reviewing. These phases are repeated throughout the development process to facilitate the rapid delivery of a

Minimum Viable Product (MVP) and the continuous improvement of system features based on user feedback (Günther et al., 2018). This approach is most appropriate for the TravelBuddy project, which emphasises responsiveness, usability and integration with third-party services such as weather APIs and Supabase.

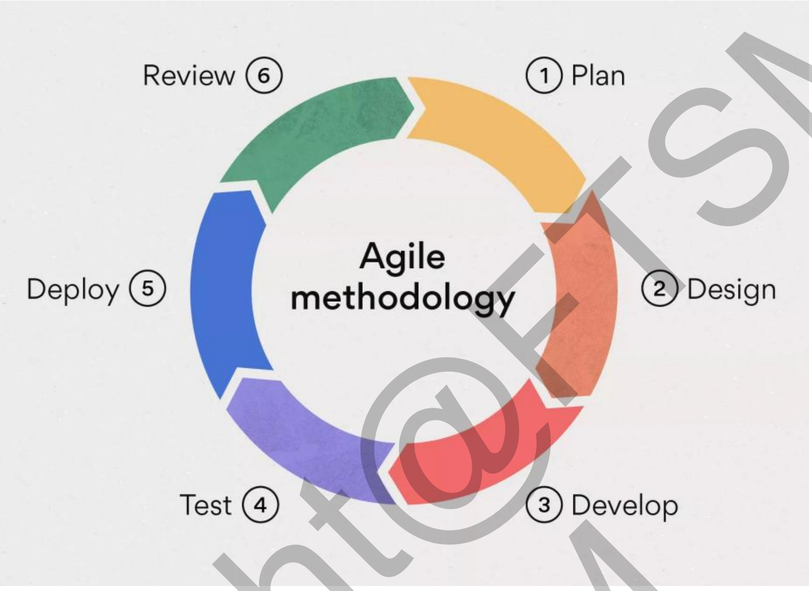

figure 1.1 Agile Development Cycle Diagram

4.1 Planning Phase

Planning started with an extensive research on the needs and expectations of users related to travel. This involved determining key features such as itinerary organization, destination suggestion, and current weather conditions. Discussions with stakeholders and literature reviews were held in order to carry out feature prioritization. The result was a feature backlog that was prioritized, defining core functionality to be implemented in initial iterations.

4.2 Design Phase

The design stage involved user interface (UI) and database schema designing. Figma and similar tools were utilized to prototype critical pages such as login, registration, homepage, and itinerary planner. A mobile-first, responsive design with simple navigation was the focus. Database designing was represented using ER diagrams and subsequently implemented in Supabase with PostgreSQL. This provided data structure consistency and optimized fetching for itinerary, place, and user management modules.

4.3 Development Phase

The development was conducted in iterative sprints with Visual Studio Code as the primary IDE. The frontend was built on Vue.js with Vue Router, and the backend logic was substituted with Supabase functions and Auth API. The Agile cycle enabled flexible deployment of the code, modular integration, and real-time testing after every sprint. External libraries like Axios, Day.js, and OpenWeatherMap API were incorporated to increase functionality.

4.4 Testing phase

In keeping with Agile principles, testing was carried out continuously throughout the development process. Unit tests and manual exploratory testing were conducted after every sprint. User acceptance testing (UAT) was conducted with a small group of users to evaluate the usability and satisfaction of features such as 'Add to Itinerary' and 'Favourite Places'. Any bugs were recorded and fixed in later sprints. Each module was tested for data consistency, validation and real-time interaction.

4.5 Deployment and review

The system was deployed live on a staging setup via Vercel (front end) and Supabase (back end). Once a stable MVP had been reached, the system was assessed by the developer and academic supervisor. UAT and supervisor feedback suggestions were tracked and utilised to enhance the UX design and feature development. The review stage also involved verifying performance across devices and browsers.

4.6 Data collection

Primary research was carried out through Google Forms circulated among travellers and university students. The survey collected data on travel planning behaviour, online behaviour and feature priorities. The feedback indicated a strong preference among users for centralised itinerary features and personalised suggestions, which validated the project's design direction. Survey feedback also informed UX enhancements such as more intuitive navigation and more prominent category filters.

5. SYSTEM DESIGN

This section details the design of the TravelBuddy system, including the database schema, system modules, and front-end interfaces created during the system development life cycle.

5.1 User Requirement Analysis

The system was modeled following the analysis of user requirements, which showed major features needed for an intelligent travel planning system. They are user authentication, attraction finding, itinerary personalization, and comment management. The modules that were implemented are as follows:

1.User Login and Registration
Users sign up and log in with Supabase Auth, which offers secure, scalable identity management (Supabase, 2024). User metadata is saved in a profiles table, with the option to enrich it with full name, avatar, and personal bio.

2.Home Dashboard
On logging in, the users are taken to the Home page that showcases suggested destinations, weather forecasts (through OpenWeatherMap API), and navigation features.

3.Attractions and Sub-Attractions
Users can browse attractions saved in the places table, along with metadata like category, rating, and images. Each place has several sub-attractions (sub_attractions table) to enable finer-grained travel planning.

5. Comment and Review Section
Users are able to read and write comments on places through the comments table, adding interactivity and social proof functionality.

5.Itinerary Planning
Users can select places, assign dates, and create personalized travel plans using the itineraries and activity_plans tables.

6.Favorites
The favorites table holds locations bookmarked by users for future reference.

7.Admin Management An isolated admins table provides privileged access to manage data, allowing moderation of users, attractions, comments, and activity plans.

5.2 Database Development

The backend database was constructed with Supabase PostgreSQL, a contemporary, scalable, open-source relational database management system that is friendly to real-time applications (Supabase, 2024). The model is based on relational modeling and normalization principles (Coronel & Morris, 2019).

As shown in Figure 5.1, the database schema comprises:

1.users (via Supabase Auth): Handles authentication.

2.profiles: Extends user data with editable attributes.

3.places: Stores all travel destinations.

4.sub_attractions: Stores smaller points of interest under each place.

5.itineraries: Links users to travel plans.

6.activity_plans: Breaks itineraries into detailed time slots.

7.favorites: Tracks user bookmarks.

8.comments: Captures user reviews per place.

9.admins: Secures privileged admin access.

10.selected_places: Stores places pre-selected for itinerary planning.

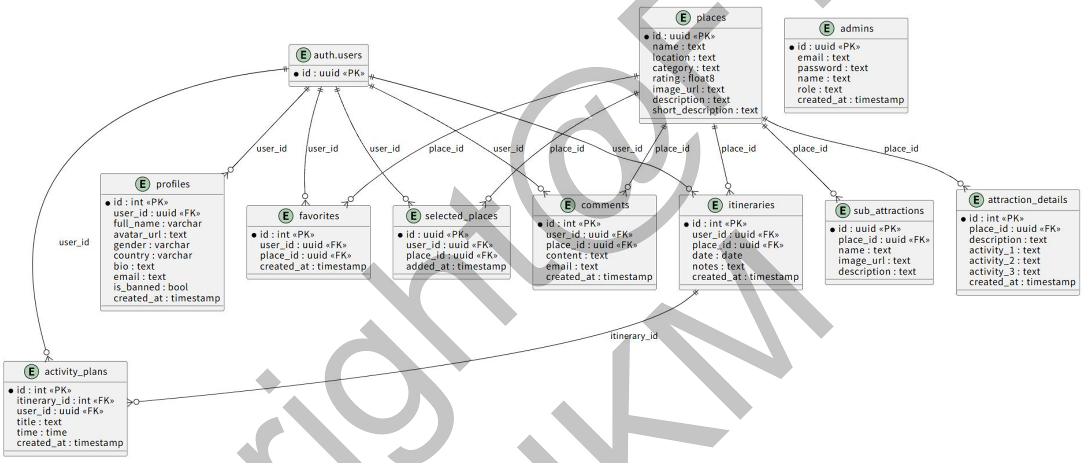

Figure 5.1: Entity-Relationship Diagram of TravelBuddy Database

5.3 Interface and Module Implementation

The user interface was created using Vue.js 3, a progressive JavaScript framework designed for modular, reactive and component-based development (Vue.js, 2023). It enables responsive design on desktop and mobile platforms.

Each page is realised as a separate component:

- Login.vue / Register.vue: Authentication
- Home.vue: Main dashboard
- Recommendation.vue: Attraction list
- AttractionDetail.vue: Detailed place info with comments
- Itinerary.vue: Travel planner
- Profile.vue: Editable user info and favorites
- AdminPanel.vue: Centralized admin interface

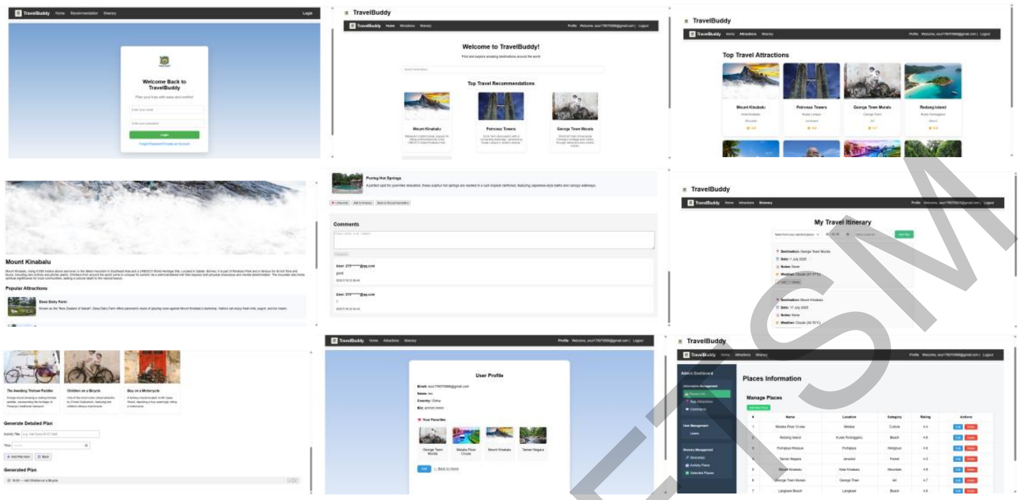

Figure 5.2: Screenshots of TravelBuddy Interface – Home, Details, Profile, and Admin Pages

6. CONCLUSION

In summary, the TravelBuddy system has been successfully developed and deployed as an online travel planning and recommendation tool. Despite some minor issues, the system has achieved its core objectives, enabling users to discover attractions, plan personalized itineraries, and easily manage their travel plans.

Using Supabase as the backend service provider and Vue.js for the frontend has enabled us to create a modular, scalable and responsive system architecture. Major functionalities such as attraction browsing, user registration, personalised itineraries and admin-level data management have been fully implemented and tested.

In addition, the system provides users with an engaging and informative experience, offering features such as real-time commenting, sub-attraction discovery and a versatile itinerary planner backed by weather data integration. While there is scope for future growth, e.g. in the form of a mobile-first design and refined recommendation algorithms, TravelBuddy provides a solid foundation for the ongoing research and development of intelligent travel systems.

It is expected that this system not only meets the immediate requirements of end users but also serves well as a guide for future academic and business development in web-based travel planning systems.

ACKNOWLEDGEMENT

During the course of my research, I would like to thank my supervisor, Dr. Zaihosnita Hood, for her insightful guidance, fruitful feedback, and unwavering support throughout the course of this project. Her advice and encouragement have been invaluable in shaping the scope and quality of my research.

I also want to thank the Faculty of Science and Technology, Universiti Kebangsaan Malaysia, for permitting the usage of the facilities, technical resources, and use of research facilities that enabled the success of this project.

Special thanks go to my seniors and classmates who provided me with much valuable advice and assistance during the research process. Their friendship and support have benefited me greatly in pursuing my degree.

My heartfelt thanks to my family, friends, and colleagues for their strong support, motivation, and understanding throughout this journey. They have been my pillar of strength and motivation all along.

I would like to thank all the people who have contributed directly or indirectly towards the completion of this research.

REFERENCE

Buhalis, D., & Law, R. (2008). Progress in information technology and tourism management: 20 years on and 10 years after the Internet. Tourism Management, 29(4), 609-623.

Xiang, Z., Wang, D., O'Leary, J. T., & Fesenmaier, D. R. (2015). Adapting to the digital marketplace: How the internet is changing the competitive structure of tourism. Journal of Travel Research, 54(4), 563-577.

Neuhofer, B., Buhalis, D., & Ladkin, A. (2015). Smart technologies for personalized experiences: A case study in the hospitality domain. Electronic Markets, 25(3), 243–254.

Gretzel, U., Sigala, M., Xiang, Z., & Koo, C. (2015). Smart tourism: Foundations and developments. Electronic Markets, 25(3), 179–188.

Sommerville, I. (2016). Software Engineering (10th ed.). Pearson.

Gaurav, H., & Pradeep, S. (2012). Agile Methodology for Mobile Application Development: A Comparative Study. International Journal of Computer Applications.

Günther, K., Ross, B., & Jäger, A. (2018). Developing MVPs in Agile Projects. Journal of Software Development.

Google Forms. (2025). Travel Planning User Research Survey. [Online survey data].

Supabase. (2024). Supabase Documentation. Retrieved from https://supabase.com/docs

Coronel, C., & Morris, S. (2019). Database Systems: Design, Implementation, and Management (13th ed.). Cengage Learning.

Vue.js. (2023). Introduction to Vue.js. Retrieved from https://vuejs.org/

OpenWeather. (2024). Weather API. Retrieved from https://openweathermap.org/api

LI MINGJIE(A191284)
Dr.ZAIHOSNITA HOOD
Faculty of Information Science & Technology
The National University of Malaysia