**Purpose of the software includes**

#### \*\*1.

Types of software development processes for applications\*\*

This project adopts Agile as the core development process and implements iterative development based on the Scrum framework.

The team conducts a Sprint every two weeks to ensure rapid delivery and validation of functional modules such as itinerary generation, map integration, and user preference analysis.

---

#### \*\*2.

Reason for choosing this type\*\*

The core reasons for choosing agile development are as follows:

-Dynamic demand and user feedback driven\*\*

According to the 2024 Global Self Travel Trends Report, 78% of users hope to adjust their routes in real-time during their travel planning (data source:

Statista）。

Agile development allows for prioritizing the development of high demand features (such as "intelligent congestion avoidance algorithms") in each Sprint and quickly optimizing them through user testing.

-\* \* Technical exploration and risk control\*\*

Travel planning involves integrating multiple data sources (such as Google Maps API, weather data interface), and agile development supports phased verification of technical feasibility.

For example:

-Case 1: In the first Sprint, the team verified the stability of the attraction's opening time API and found that some interfaces had high latency. They promptly switched to a local caching solution.

-Transparency in team collaboration\*\*

Use Jira for task tracking and conduct daily stand ups to ensure clear division of labor among members.

For example:

-Case 2: Backend developers discovered performance issues with map rendering in Sprint Review and used collaborative optimization algorithms to reduce loading time from 5 seconds to 1 second.

---

#### \*\*3.

Possible uses of software (target market)\*\*

This software focuses on personalized travel planning. The target users and market analysis are as follows:

#####\* \* Core functions and usage scenarios\*\*

-Intelligent itinerary generation\*\*

-Case 3: When a user inputs "3-day parent-child trip to Macau", the app automatically recommends:

-Day 1: Macau Science Museum (Children's Interactive Area) → Macau Tower Sightseeing (Family Ticket Discount) → Black Sand Beach Picnic.

-Day 2: Luhuan Fishing Village (cultural experience) → Guanye Street food exploration (allergen screening function).

-\* Data support \*: According to Ctrip's "2023 Parent Child Travel Report", 63% of household users hope that their itinerary includes attractions that balance "education+entertainment".

-\* \* Dynamic adjustment and warning\*\*

-\* Case 4 \*: The user originally planned to visit the memorial archway of Dasanba, and the APP detected that the real-time crowding degree reached 90% (data source: tourist flow monitoring API in the scenic spot), and automatically pushed the alternatives:

-Alternative 1: Visit the Macau Museum in the morning (with a 30% foot traffic), and then head to Taipa Lung Wan Portuguese Charm in the afternoon.

-Alternative 2: Enable "off peak mode" and recommend visiting in the evening to avoid peak hours.

#####\* \* Target market segmentation and data support\*\*

-\* \* Individual travelers (aged 18-35)\*\*

-Market size: According to Macau Tourism Office data, the proportion of independent travelers will reach 65% in 2023, a year-on-year increase of 12%.

-Pain point: 72% of users believe that existing tools (such as TripAdvisor) lack dynamic adjustment capabilities (source: user survey questionnaire).

-\* \* Family users\*\*

-Requirement analysis: 85% of household users hope that their itinerary includes "child friendly facilities" and "rest points" (source: Tongcheng Travel's "Family Travel White Paper").

-\* \* Business travelers\*\*

-Efficiency requirement: The average travel planning time for business users needs to be reduced from 2 hours to 20 minutes (source: internal MVP test feedback).

#####\* \* Competitive advantage\*\*

-Deep personalization algorithm\*\*

Using Collaborative Filtering and Genetic Algorithm to optimize the route:

-Case 5: Based on 1000 historical travel data, the algorithm increased user satisfaction from 3.8/5 to 4.5/5 (test dataset).

-Offline mode and low network dependency\*\*

Support local caching of maps and scenic spot information to solve network coverage issues in remote areas such as Coloane Country Park.

**Software development plan**

**Development Process**

**Needs Analysis:**

Conduct user research to understand user needs and pain points.

Conduct competitive analysis to identify market trends and functional differences.

Prioritise features based on research results to ensure that core features are prioritised for development.

**System Design:**

Define technology stack and tools for system architecture design.

Draw database ER diagram and define data model.

Design API interfaces to ensure effective communication between front and back end.

Create UI prototype diagrams for user interface design review.

**Core development:**

Split development by module to ensure each module is independent and testable.

Develop core functionality such as recommendation algorithms, route planning and booking systems.

Write unit tests to ensure code quality and functional correctness.

**Integration Testing:**

Perform multi-module co-tuning to ensure that the modules work together.

Perform performance pressure testing to ensure system stability under high load.

Verify compatibility with third-party APIs to ensure system scalability.

**Deployment Go-live:**

Deploy the application in the cloud environment to ensure high availability and scalability.

Build monitoring system to monitor system health in real time.

Publish in App Store and Google Play to ensure users can download and use.

**Maintain iterations:**

Collect user feedback, conduct regular requirement reviews and feature optimisation.

Continuously update security patches to ensure system security and stability.

**Members (Roles & Responsibilities & Portion)**

**Product Manager:**

Requirements gathering: Understand users' requirements for travel apps through user surveys, e.g. expectations for recommended itineraries.

Prioritisation: Based on user feedback, the ‘real-time flight information’ feature was given high priority.

User communication: Conduct regular interviews with users to understand their experience and suggestions for improvement.

Product roadmap: Plan for future feature releases, e.g. ‘social sharing next quarter’.

**Developers:**

Module development: Implemented the ‘hotel search’ feature to ensure that users could quickly find the right accommodation based on date and location. Implemented a recommendation algorithm to suggest suitable places to visit based on user preferences, historical data and popular attractions. Responsible for the development of a route planning module that uses mapping APIs (e.g. Google Maps API) to calculate the best route from one attraction to another.

Code Implementation: Developed the backend using Java and Django frameworks to ensure the efficiency of the system.

**Testers:**

Integration testing: testing the hotel booking and payment functions work together to ensure a smooth user experience.

Performance testing: Simulate high concurrency user access to ensure that the App can still respond quickly during peak hours.

Bug Reporting: Use bug tracking tools to record issues found and communicate with the development team on the progress of fixing them.

**UI/UX Designer:**

User Journey Mapping: Mapping the user path through the App to identify optimisation points.

UX Optimisation: Iterate design based on user feedback to improve interface friendliness and functional ease of use.

**Schedule**

**1. Needs analysis**

Duration: approximately 1 week

Main activities:

User research and interviews

Competitor analysis

Function prioritisation

**2. System Design**

Duration: about 1 week

Main activities:

Determine technology selection and architecture design

**3. Core Development**

Duration: about 3 weeks

Main activities:

Module development (recommendation algorithm, route planning, booking system)

Write and execute unit tests

Complete integration of core functionality

Submission of runnable core module code and unit test reports

**4. Integration testing**

Duration: about 1 week

Main activities:

Multi-module debugging

Performance pressure testing and stability verification

Third-party API compatibility testing

**5. Deployment on-line**

Duration: about 1 week

Main Activities:

Cloud environment deployment

Monitoring system setup

App Store and Google Play release

**6. Maintenance iterations**

Duration: Ongoing

Key Activities:

User feedback collection (monthly evaluation)

Collect feedback through user surveys, questionnaires, etc.

Functional optimisation (every 1-2 months)

Iterate and optimise features based on feedback

Security patch updates (ongoing)

Regularly check and update system security

New feature iteration (every 2-3 months)

Plan and develop new features based on market demand

Regular version updates (every 2 months)

Release new versions with optimisations and new features

**Total development time: ~7 weeks (excluding ongoing maintenance iterations).**

**Algorithm**

**1. Recommend personalised attractions based on user preferences and behaviour.**

Data Collection:

Collect user preference data (e.g., interests, past visits).

Collect attraction data (e.g., popularity, ratings).

Data Processing:

Normalise and pre-process data.

Implement the following algorithms:

Collaborative filtering: recommending attractions based on similar users.

Content-based filtering: recommending attractions based on their features.

**2. calculate the best route between the selected attractions.**

Input Collection:

Get the attractions and starting points selected by the user.

Implement the following algorithms:

Dijkstra's algorithm: for shortest paths in weighted graphs.

A\* search algorithm: for more complex path planning with heuristics.

Route Optimisation:

Considers factors such as traffic conditions and road closures.

Provides multiple route options (fastest, scenic, etc.).

Output:

Displays the recommended route on a map with step-by-step navigation.

**3. Help users choose the right hotel based on user reviews and ratings and based on the user's budget, recommend hotels within the appropriate price range.**

Data Collection:

Collect user ratings and reviews of hotels.

Tally user feedback (e.g., cleanliness, service, location, etc.).

Collect user's budget range (e.g., lowest and highest price).

Implement the following algorithm:

Generate a list of recommended hotels based on the composite score.

Can be filtered based on user preferences (e.g., amenities, location).

Get the price information of the hotels from the database and filter the hotels that meet the user's budget.

Compare the prices of the selected hotels, taking into account promotional and discount information.

**Current status of your software**

Work completed

**1.Requirements Analysis:**

Collect user requirements through questionnaires and user interviews.

Define target user groups and their usage scenarios.

Define core functional modules, such as recommendation system, route planning, hotel booking, etc.

**2.System Design:**

Complete the system architecture design, including front-end and back-end technology selection.

Design the database structure to ensure efficient data storage and retrieval.

Develop API interface specifications to facilitate front-end and back-end interaction.

Ongoing Work

Core function development:

Personalised recommendation system:

Use machine learning algorithms to analyse user behaviour and preferences.

Implement preliminary versions of collaborative filtering and content recommendation algorithms.

**3.Route planning functionality:**

Integrate map service APIs to calculate optimal routes.

Develop user interface to allow users to customise route settings.

Hotel booking module:

Access third-party hotel data APIs for real-time price and review information.

Develop hotel search and filtering function.

User Testing

Plan Implementation

**1.Internal testing:**

After the development of core functions is completed, conduct functional testing and stress testing.

Organise team members to conduct initial experience and record feedback.

**2.User feedback collection:**

Create feedback forms for test users to provide experience and suggestions for improvement.

Hold regular meetings to analyse user feedback and develop iteration plans.

Iterative Development:

Based on user feedback, prioritise fixing key issues and optimising user experience.

Conduct version iteration to gradually improve the function.

Market Research

Continuous attention

**1.User demand analysis:**

Conduct regular market research to understand user preferences and emerging travel trends.

Monitor competitors' product updates and market strategies.

Technical Updates

Regular maintenance

**1.Algorithm Updates:**

Regularly assess the accuracy of recommendation algorithms for optimisation and improvement.

Keep an eye on emerging technologies and algorithms to assess their applicability in the application.

**2.Technology Stack Updates:**

Regularly update development frameworks and libraries to ensure that the latest security patches and features are used.

Conduct code reviews to optimise code quality and performance.

**Future plan**

**1. Functionality Expansion**

1.1 New Features

Multi-language support:

Add multiple language options to expand the user base.

Social features:

Users can share their trips and travel experiences to enhance community interaction.

Offline Mode:

Provide offline maps and trip information, convenient for use in a network-less environment.

1.2 Intelligent Recommendation System

Deep learning algorithm:

Deep learning technology is used to improve the accuracy of personalised recommendations.

Real-time data integration:

Integrate real-time weather, traffic and activity information to optimise trip arrangements.

**2. User Experience Optimisation**

2.1 User Interface Improvement

Interface beautification:

Update the design style to make the interface more modern and friendly.

Interaction experience enhancement:

Optimise the user interaction flow and reduce the operation steps.

2.2 User feedback mechanism

Regular survey:

Regularly collect user feedback to understand the use of pain points and changes in demand.

User support:

Add online customer service function to quickly respond to user problems.

**3 Marketing**

3.1 Marketing Strategy

Social media promotion:

Carry out promotional activities through social media platforms to attract more users.

Partnership:

Establish partnerships with travel companies, airlines and hotels to offer deals and packages.

3.2 User Growth

Incentive Mechanism:

Design an invitation incentive mechanism to encourage users to invite their friends to use the app.

User Loyalty Programme:

Introduce a point system to encourage users to use frequently.

**4. Technical Updates**

System Optimisation

4.1 Performance Monitoring:

Regularly monitor system performance to ensure high availability and fast response.

4.2 Security Upgrade:

Enhance data security measures to protect user privacy and information security.

---