TRAVEL GUIDES AND SERVICE WEBSITE

ZHANG ZHIHANG，Shahrul Azman Mohd Noah

Faculty of Information Science & Technology

Universiti Kebangsaan Malaysia

43600 Bangi, Selangor

Abstract

Fokus projek ini adalah untuk membangunkan laman web panduan perjalanan dan perkhidmatan, bertujuan untuk menyediakan platform bagi pengguna untuk menerbitkan panduan perjalanan dan bertukar pengalaman perjalanan antara satu sama lain. Pada masa ini, laman web pelancongan arus perdana terutamanya mengesyorkan tarikan dan laluan pelancong serta kekurangan pengalaman pengguna sebenar dan ciri interaktif, menyukarkan pengguna untuk mencari maklumat perjalanan yang sepadan dengan minat mereka. Untuk menyelesaikan masalah ini, berdasarkan laluan perjalanan yang disyorkan, mencipta lebih tertumpu pada pengalaman sebenar pengguna, anda boleh mendapatkan pelbagai kandungan yang kaya dengan laman web petua perjalanan. Metodologi penyelidikan meliputi keperluan, reka bentuk, fasa pelaksanaan dan ujian. Kami menjalankan ujian kebolehgunaan ke atas 10 pengguna, termasuk pelajar dari China dan Malaysia. Keputusan ujian menunjukkan bahawa kebanyakan pengguna meluluskan reka bentuk dan pembangunan laman web.

Abstract

The focus of this project is to develop a travel guides and services website, aiming to provide a platform for users to publish travel guides and exchange travel experiences with each other. Currently, mainstream travel websites mainly recommend tourist attractions and routes and lack real user experience and interactive features, making it difficult for users to find travel information that matches their interests. To solve this problem, on the basis of recommended travel routes, create a more focused on the user's actual experience, you can get a variety of content-rich travel tips website. The research methodology covers the requirements, design, implementation and testing phases. We conducted a usability test on 10 users, including students from China and Malaysia. The test results showed that most users approved of the design and development of the website.

1.0 INTRODUCTION

In recent times, the landscape of international tourism has undergone a profound transformation, with a significant surge in global travel activities as more individuals venture to diverse tourist destinations across the world (Şahin, 2012, p1). This growth, however, is accompanied by escalating competition in the tourist marketplace, making the enhancement of customer service quality a pivotal factor in attracting and retaining visitors to specific destinations. Amidst this dynamic, the demand for a personalized, efficient, and integrated travel platform has become increasingly prominent—one that empowers users to access comprehensive travel guides, exchange authentic experiences, and book services seamlessly. Such a platform is not merely a convenience but a necessity to address the evolving needs of modern travelers (Aram Izzeddin Nejmeddin, 2020).

This project, centered on developing a Travel Guides and Service Website, is rooted in both practical relevance and substantial social impact. Today's travelers frequently grapple with a series of frustrations: information scattered across disjointed sources, from outdated guidebooks to inconsistent online reviews, leaving them uncertain about the reliability of insights; a dearth of genuine, user-generated content that captures the nuances of local culture, hidden gems, and real-time experiences; and cumbersome booking processes that involve navigating multiple websites or apps, often leading to confusion and inefficiency. These challenges not only hinder the joy of travel planning but also limit opportunities for meaningful cultural engagement.

By creating a platform that integrates comprehensive travel guides, a vibrant community for experience sharing, and a streamlined booking system, this project aims to redefine the travel planning journey. It seeks to simplify the logistical complexities of travel, allowing users to curate itineraries tailored to their preferences—whether they crave adventure, culinary exploration, or cultural immersion. Beyond convenience, the platform fosters a community-driven approach, where travelers from diverse backgrounds can connect, share stories, and exchange tips, thereby promoting cross-cultural understanding and authentic interactions with local communities. In doing so, it not only enhances the quality of individual travel experiences but also contributes to the sustainable development of tourism by encouraging respectful and informed engagement with destinations. Ultimately, this website aspires to be more than a tool—it aims to be a gateway that enriches the way people explore, connect, and experience the world.

2.0 LITERATURE REVIEW

In recent times, there has been an increase in international tourism activities with more people travelling to different tourist destinations across the world (Şahin, 2012, p1). Importance of improving the level of customer service quality for achieving increased number of tourists to a particular tourist destination has gained momentum in the wake of competitiveness of

tourist marketplace. Among them, a personalized, efficient and integrated travel platform that enables users to access comprehensive travel guides, share experiences and book services seamlessly is very important to meet the needs of travelers (Aram Izzeddin Nejmeddin, 2020).

This topic was chosen for its practical relevance and potential social impact. Travelers often encounter challenges such as fragmented information sources, lack of authentic user-generated insights and cumbersome booking processes. By developing a platform that integrates these features, the project not only simplifies travel planning, but also promotes a community-driven approach that encourages cultural exchange and authentic travel experiences.

Now users' opinions on travel websites are very important. Today, a majority of travelers expressing preference for AI driven recommendations. Efficiency in operations, manifested through tools like chatbots and predictive analytics, emerges as a pivotal advantage. Virtual Reality (VR) and drones are highlighted as transformative tools, offering tourists novel ways to explore destinations (T. Milton, 2023).

3.0 METHODOLOGY

This section introduces the complete system design and implementation of a travel guide and service website developed with a front-end and back-end separation architecture. It aims to help users understand the purpose, structure, functional modules and development process of the system. The system aims to provide users with services such as browsing travel guides, viewing recommended travel routes, submitting feedback and placing orders. In terms of testing, white box testing and User Acceptance testing are designed to test the code logic operation and user satisfaction with various aspects of the system respectively.

3.1 User Needs

Recommended Tourist Attractions and Routes

Recommend popular attractions and the best travel routes. Provide detailed information about attractions, including ticket prices, opening hours, tourist reviews, etc. Combined with the map function, understand the local traffic conditions.

Publish and Browse Travel Guides

Users can write and publish their own travel guides, travel notes, and experience sharing. Other users can like, comment, and collect high-quality guides to form interactive exchanges. The system automatically recommends high-quality content to help new users quickly obtain useful travel information. Provide multimedia support, and upload pictures and videos to make the content more intuitive and vivid.

4.1 System Model

The system model section outlines the object-oriented approach used to build the project. The focus of this section is to represent the interaction, processes and functional requirements between visualization and the system through diagrams (Figures 1,2,3,4,5).

3.2.1 Case diagram

Use case diagram showing the interaction between administrator, user and system.

Role Analysis:

Administrator: Mainly responsible for managing system content, including orders, products, travel guides, reviews, announcements, etc. Has permissions such as Audit, Manage, Reply, etc.

User: Mainly responsible for browsing and submitting content, such as submitting questions, browsing announcements, purchasing products, etc.

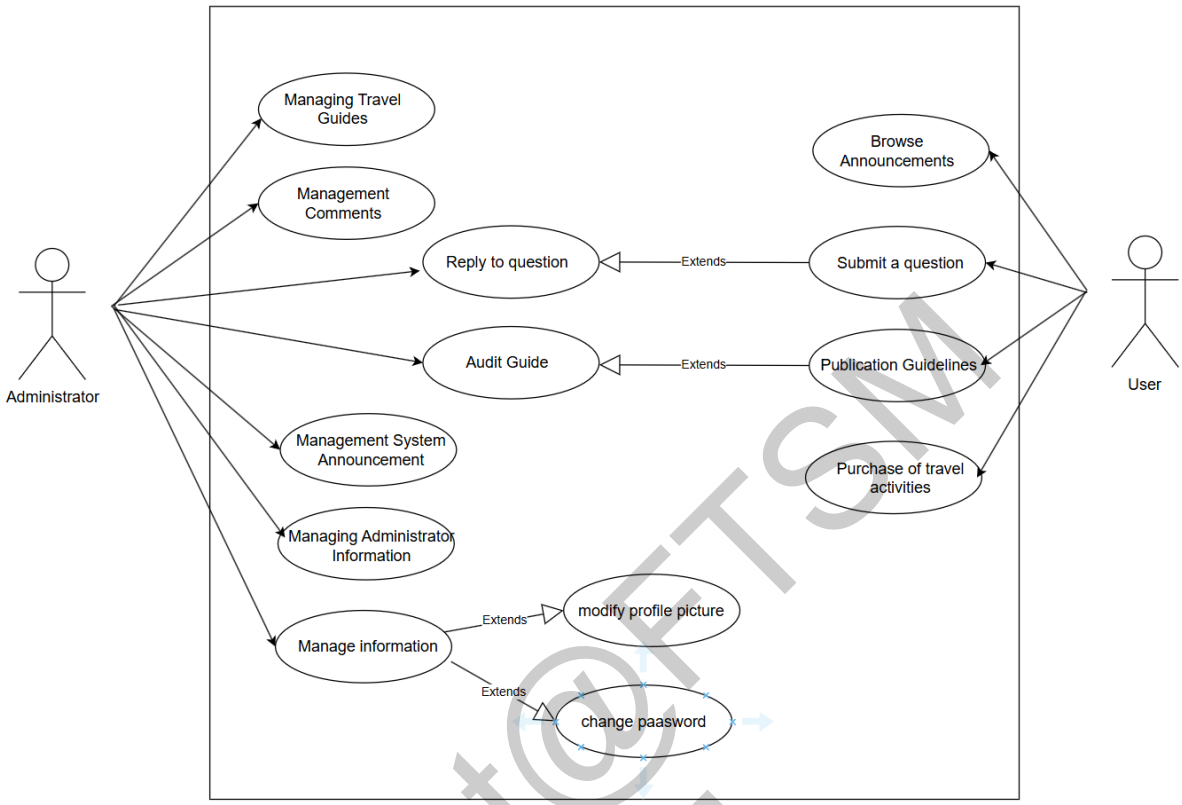


Figure 1 Functional requirements of the system

3.2.2 Sequence diagram

Sequence diagrams: These diagrams show the sequence of interactions between users and the system during key processes.

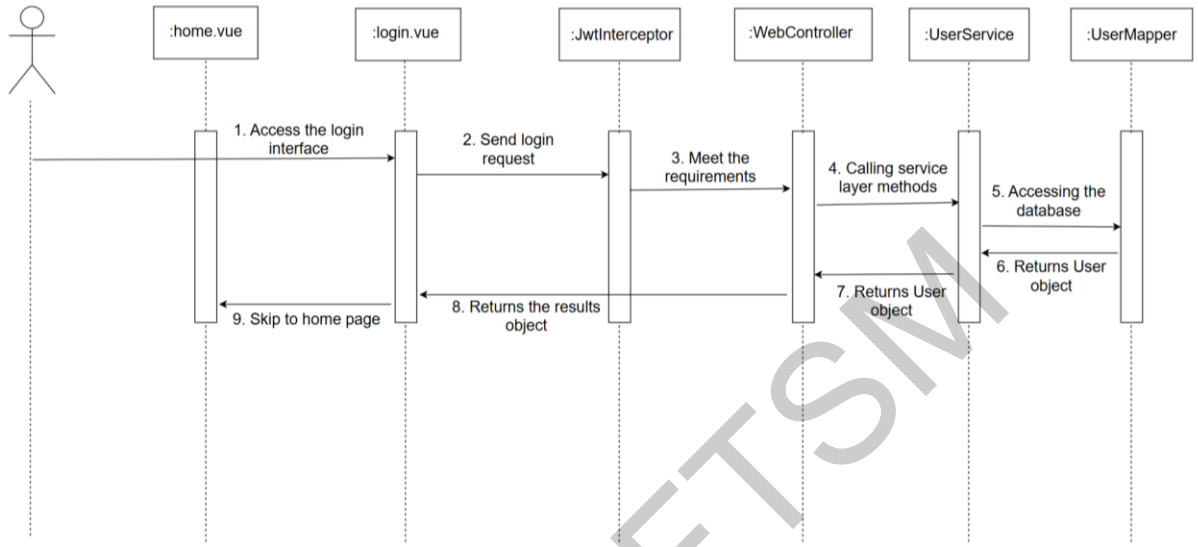


Figure 2 User Login Sequence Diagram

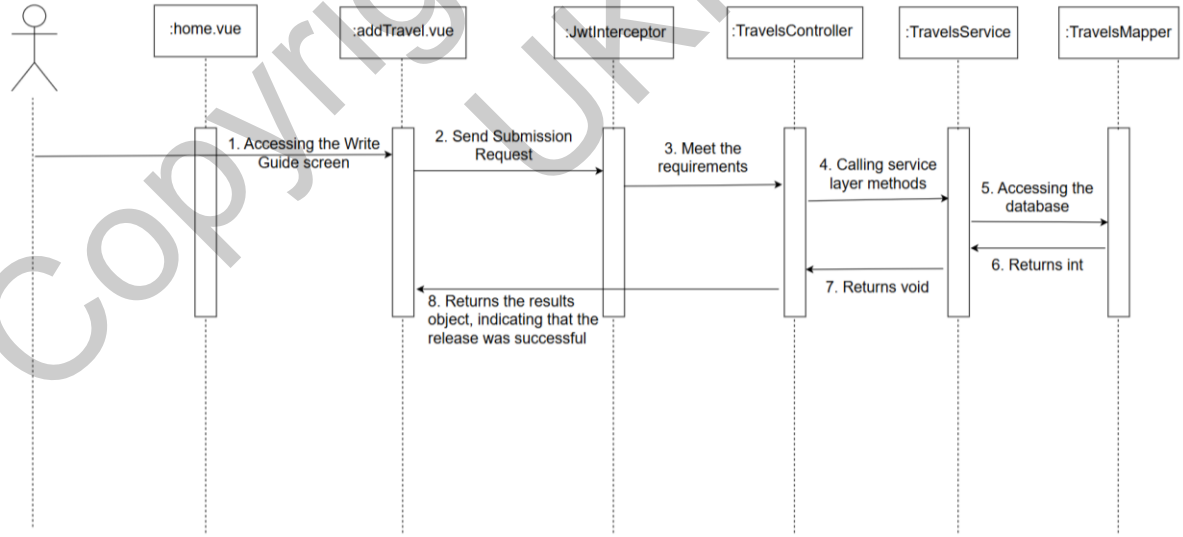


Figure 3 User Publishing Travel Guide Sequence Diagram

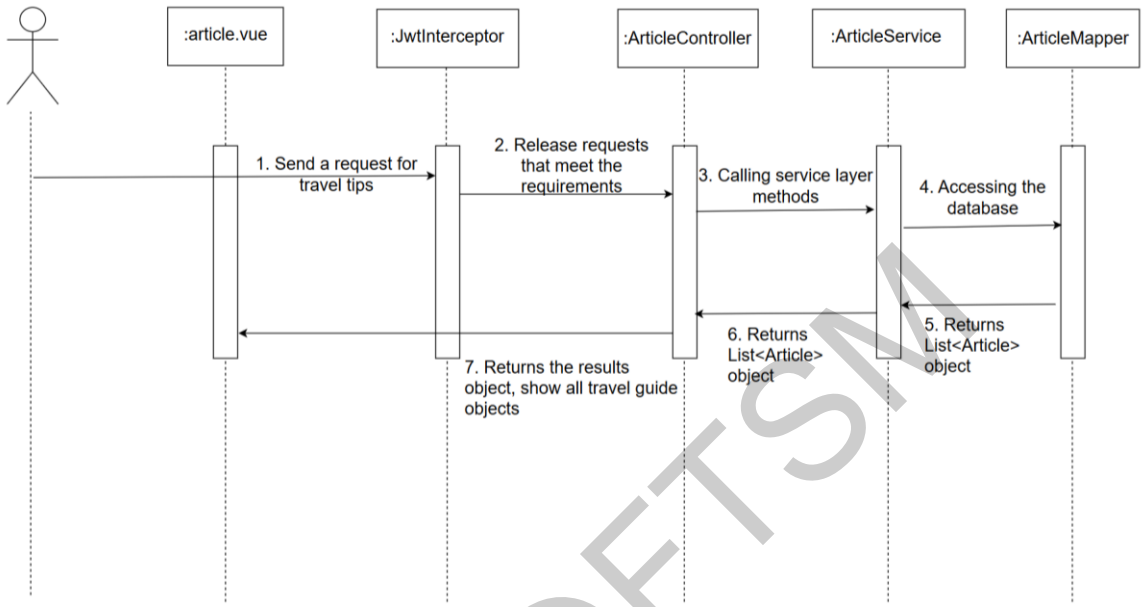


Figure 4 User Browse Travel Guides Sequence Diagram

3.2.3 System Architecture Diagram

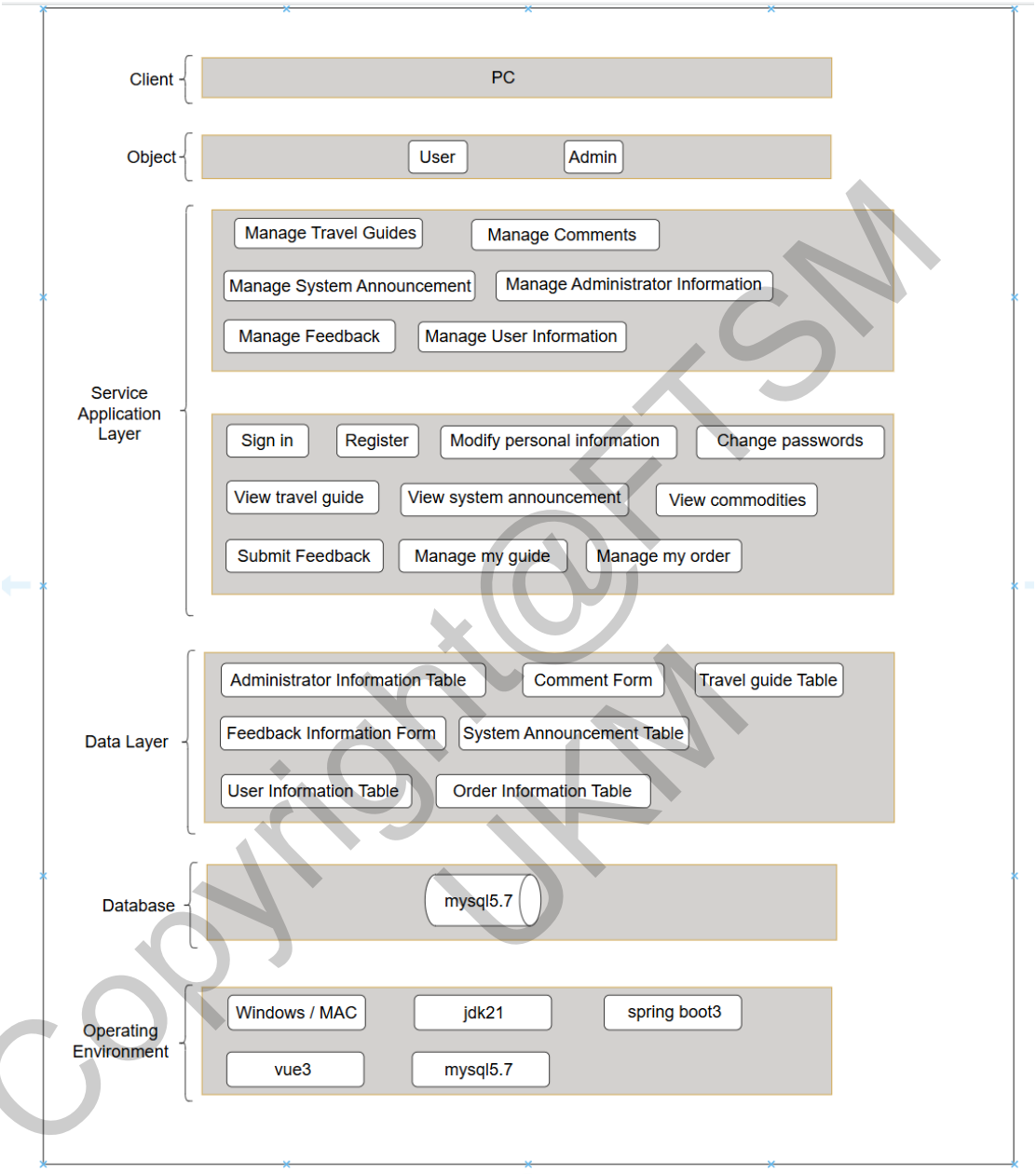


Figure 5 System Architecture Diagram

4.0 RESULTS

4.1 SYSTEM COMPONENTS

i. User Login

This module consists of two parts, divided into normal users and administrators. All users enter a username and password. After submitting the login form, the system will use JWT to verify the credentials and roles through the backend authentication. The token contains the user ID and role information. If correct, the system will grant access to the corresponding interface. Otherwise, the user will receive an error message and be denied access.

Clearly define the user role as USER for backend verification.

```
form: { role: 'USER' },
```

Figure 6 Define the user role

After the user successfully logs in, the user information (including token) is cached in localStorage.

```
if (valid) { // 表示表单校验通过
  request.post( url: '/login', data.form).then(res => {
    if (res.code === '200') {
      ElMessage.success( options: 'Login Successful')
      // 存储用户信息到浏览器的缓存
      localStorage.setItem('xm-user', JSON.stringify(res.data))
      router.push('/front/home')
    } else {
      ElMessage.error(res.msg)
    }
  })
}
```

Figure 7 Validate the login form and send request

Automatically jump to the front page after successful login.

Extract the user ID and role fields in the Token to ensure that the front-end user identity is valid

```
String audience = JWT.decode(token).getAudience().get(0);
String userId = audience.split( regex: "-")[0];
String role = audience.split( regex: "-")[1];
// 根据用户角色判断用户属于哪个数据库表 然后查询用户数据
if (RoleEnum.ADMIN.name().equals(role)) {
    account = adminService.selectById(Integer.valueOf(userId));
} else if (RoleEnum.USER.name().equals(role)) {
    account = userService.selectById(Integer.valueOf(userId));
}
```

Figure 8 TokenUtils – get current user from JWT

ii. User writing travel guides

Users can click the Writing Travel Guides button to publish their travel guides. This function is mainly used for users to submit travel log information, upload cover pictures, and enter departure time, travel expenses, locations and days. After the administrator reviews and approves it, it can be displayed on the homepage.

```
<div>
  <el-form ref="form" :model="data.form" label-width="90px" style="..." :rules="data.rules">
    <el-form-item prop="title" label="Title">
      <el-input v-model="data.form.title" placeholder="Enter Title"></el-input>
    </el-form-item>
    <el-form-item label="Cover">
      <el-upload
          :action="baseUrl + '/files/upload'"
          :headers="{ 'token': data.user.token }"
          list-type="picture"
          :on-success="handleFileUpload"
      >
        <el-button type="primary">Upload Photos</el-button>
      </el-upload>
    </el-form-item>
    <el-form-item prop="descr" label="Description">
      <el-input :rows="3" type="textarea" v-model="data.form.descr" placeholder="Enter Description"></el-input>
    </el-form-item>
    <el-form-item prop="startDate" label="Start Date">
      <el-date-picker style="width: 100%" type="datetime" v-model="data.form.startDate" format="YYYY-MM-DD HH:mm" value-format="YYYY-MM-DD
    </el-form-item>
    <el-form-item label="Cost" prop="money">
      <el-input-number style="..." v-model="data.form.money" placeholder="Enter cost" :min="1"></el-input-number>
    </el-form-item>
    <el-form-item prop="location" label="Location">
      <el-input v-model="data.form.location" placeholder="Enter Location"></el-input>
    </el-form-item>
    <el-form-item prop="days" label="Days">
      <el-input-number style="..." v-model="data.form.days" placeholder="Enter Days"></el-input-number>
    </el-form-item>
    <el-form-item label="Content" prop="content">
      <div style="...">
```

Figure 9 User Writing Travel Guides Form Design

```
<template #default="scope">
  <el-tag type="warning" v-if="scope.row.status === 'Awaiting review'">Awaiting review</el-tag>
  <el-tag type="success" v-if="scope.row.status === 'Pass'">Pass</el-tag>
  <el-tag type="danger" v-if="scope.row.status === 'Rejection'">Rejection</el-tag>
</template>
```

Figure 10 Travel Guides Review Status

iii. User order operations and administrator order management

Users select tourist attractions in the "Go Travel" column and fill in the information to submit the form to complete the order. After submitting the form, the system uploads the user information and order data to the backend database and displays the "To be paid" status. Users can click the button to pay or refund after payment. The backend will see the order status and handle shipment, cancellation.

```
const refund = (row) => {  显示用法
  ElMessageBox.confirm('Do you confirm the refund?? ', 'Confirm', { type: 'warning' }).then(res => {
    request.put( url: '/alipay/refund?orderNo=' + row.orderNo).then(res => {
      if (res.code === '200') {
        ElMessage.success( options: 'Operation Successful')
        data.formVisible = false
        load()
      } else {
        ElMessage.error(res.msg)
      }
    })
  }).catch(err => {})
}

const pay = (row) => {  显示用法
  let baseUrl = import.meta.env.VITE_BASE_URL
  request.get( url: '/alipay/pay?orderNo=' + row.orderNo).then(res => {
    if (res.code === '200') {
      ElMessage.success( options: 'Payment successful')
      load()
    } else {
      ElMessage.error(res.msg)
    }
  })
```

Figure 11 Payment and refund operations

```
const changeStatus = (row) => {  显示用法
  let form = JSON.parse(JSON.stringify(row))  // 防止表格数据被修改
  form.status = '已完成'
  request.put( url: '/orders/update', form).then(res => {
    if (res.code === '200') {
      ElMessage.success( options: 'Operation Successful')
      data.formVisible = false
      load()
    } else {
      ElMessage.error(res.msg)
    }
  })
}
```

Figure 12 Admin manages orders

iv. User Comment

This module enables users to comment and reply to travel notes on the platform. The comment system includes basic comment display, reply box, comment submission and deletion functions, and administrators can view and manage them in the background.

```
const addComment = (parentComment) => {  显示用法
  if (parentComment) {
    data.form.pid = parentComment.id
    data.form.content = parentComment.replyContent
  }
  if (!data.form.content) {
    ElMessage.warning( options: 'Enter Comment')
    return
  }
  data.form.fid = data.fid
  data.form.module = props.module
  request.post( url: '/comment/add', data.form).then(res => {
    if (res.code === '200') {
      data.form = {}
      ElMessage.success( options: 'Comment Success')
      load()
    } else {
      ElMessage.error(res.msg)
    }
  })
}
</script>
```

Figure 13 Comment Submission

v. User feedback

Users can enter the feedback title and content through the form and click the ‘Submit’ button to submit the feedback content to the server; the administrator can view all feedback content in a table in the background, and reply or delete each feedback, improving the maintainability of the system and user satisfaction.

```
<div style="...">😊Looking forward to your feedback</div>
<el-form ref="formRef" :model="data.form" label-width="80px" style="..." :rules="data.rules">
  <el-form-item prop="title" label="Title">
    <el-input v-model="data.form.title" placeholder="Please enter title"></el-input>
  </el-form-item>
  <el-form-item prop="content" label="Content">
    <el-input type="textarea" :rows="5" v-model="data.form.content" placeholder="Please enter content"></el-input>
  </el-form-item>
  <div style="...">
    <el-button type="primary" style="..." @click="add">Submit</el-button>
  </div>
```

Figure 14 Feedback Submission Form

```
<div class="card" style="...">
  <el-table stripe :data="data.tableData" @selection-change="handleSelectionChange">
    <el-table-column type="selection" width="55" />
    <el-table-column prop="title" label="Title" />
    <el-table-column prop="content" label="Content" />
    <el-table-column prop="userName" label="Published by" />
    <el-table-column prop="time" label="Time" />
    <el-table-column prop="reply" label="Reply" />
    <el-table-column label="Operation" width="180" fixed="right">
      <template v-slot="scope">
        <el-button type="primary" @click="handleEdit(scope.row)">Reply</el-button>
        <el-button type="danger" circle :icon="Delete" @click="del(scope.row.id)"></el-button>
      </template>
    </el-table-column>
  </el-table>
</div>
```

Figure 15 Admin Feedback Management

```
<div class="card" style="...">
  <el-table stripe :data="data.tableData" @selection-change="handleSelectionChange">
    <el-table-column type="selection" width="55" />
    <el-table-column prop="title" label="Title" />
    <el-table-column prop="content" label="Content" />
    <el-table-column prop="userName" label="Published by" />
    <el-table-column prop="time" label="Time" />
    <el-table-column prop="reply" label="Reply" />
    <el-table-column label="Operation" width="180" fixed="right">
      <template v-slot="scope">
        <el-button type="primary" @click="handleEdit(scope.row)">Reply</el-button>
        <el-button type="danger" circle :icon="Delete" @click="del(scope.row.id)"></el-button>
      </template>
    </el-table-column>
  </el-table>
</div>
```

Figure 16 Admin Feedback Management

vi. Administrator Notice Release

This module enables administrators to edit and publish notices in the background, and the notifications will be displayed on the homepage. Improved the communication efficiency and management transparency of the platform

```
<el-dialog title="Notice Information" v-model="data.formVisible" width="40%" destroy-on-close>
  <el-form ref="form" :model="data.form" label-width="70px" style="...">
    <el-form-item prop="title" label="Title">
      <el-input v-model="data.form.title" placeholder="Please Enter Title"></el-input>
    </el-form-item>
    <el-form-item prop="content" label="Content">
      <el-input type="textarea" :rows="4" v-model="data.form.content" placeholder="Please Enter Content"></el-input>
    </el-form-item>
  </el-form>
```

Figure 17 Admin Notice Publishing

.

4.2 TEST CASE DESIGN

4.2.1 User Registration

TC-REG-001 executes userMapper.insert(user), all condition judgments do not trigger exceptions, the registration logic is fully executed, and the user data is successfully inserted

TC-REG-002 Backend logic is not hit. Front-end form validation rules { required: true, message: 'Please enter account number' } intercept empty input, and back-end method is not called.

TC-REG-003 hit if (ObjectUtil.isNotNull(dbUser)) → throw new CustomException(ResultCodeEnum.USER_EXIST_ERROR); Username duplicate verification, USER_EXIST_ERROR exception, the process was terminated, and the subsequent logic was not executed.

Table 1 User Registration Test Result Table

Test ID	Test Description	Input	Expected Output	Actual Output	Pass/Fail
TC-REG-001	Correct registration	user: alice, pwd: abc12345	Registration successful, jump to login page	Registration successful, redirected to login page	Pass
TC-REG-002	Username is empty	user: "", pwd: abc12345	Prompt "Please enter your username"	Form validation error: "Please enter your username"	Pass
TC-REG-003	Duplicate Username	user: alice, pwd: abc12345	Prompt "Username already exists"	Error: "Username already exists"	Pass

4.2.2 User Login

TC-LOGIN-001 Execute UserService.login(), generate token with TokenUtils.createToken() and return user with token.

TC-LOGIN-002 triggers the password mismatch condition, throws USER_ACCOUNT_ERROR.

TC-LOGIN-003 is intercepted by front-end, backend /login not invoked.

TC-LOGIN-004 hits JWTInterceptor.preHandle() validation failure, throw TOKEN_INVALID_ERROR or TOKEN_CHECK_ERROR during decode/verify

Table 2 User Login Test Result Table

Test ID	Test Description	Input	Expected Output	Actual Output	Pass/Fail
TC-LOGIN-001	Correct login	user: user1, pwd: 123456	Login success, redirect to home, store user info	Login success, redirected and info cached	Pass
TC-LOGIN-002	Wrong password	user: user1, pwd: wrong	Error message: “Wrong username or password”	Error: “Wrong username or password”	Pass
TC-LOGIN-003	Empty form submission	user: <empty>, pwd: <empty	Prompt: “Please enter your username and password”	Validation error: “Please enter	Pass

		>		your username and password ”	
TC-LOGIN-004	Illegal JWT access	token: invalid or expired	Return TOKEN_INVALID_ERROR or TOKEN_CHECK_ERROR	JWT validation error thrown	Pass

4.2.3 Travel Guides Writing

TC-JOURNAL-001 calls TravelsService.add(), all logic is executed normally, travel data is inserted into the database, and the process is complete.

TC-JOURNAL-002 The interception point was in the rules of the front-end el-form, the back-end add() method was not triggered, and TravelsService.add() was not called.

TC-JOURNAL-003 hit TravelsService.updateById(), the travel notes status is updated successfully, and the content is refreshed.

Table 3 Travel Guides Writing Test Result Table

Test Case ID	Test Step Description	Test Data	Expected Result	Actual Result	Pass/Fail
TC-JOURNAL-001	Enter the title and content and click publish	title: "Trip to Huangshan"content: "A beautiful journey"	Travel guide uploaded successfully, "Operation	The travel guide is saved successfully, the status is "pending review", and the front-end	Pass

			successful " message shown	prompts "operation successful"	
TC-JOURNAL-002	Click publish without entering required fields	title: nullcontent: null	Form validation error: "Please enter a title" or "Please enter content"	The front-end form prompts "Please enter title/content", but travelsService.add() is not triggered	Pass
TC-JOURNAL-003	Modify existing journal and click save	Before: content: "abc"After: "abc updated"	Travel guide updated, database content modified	The update was successful, the travel journal status was updated to "pending review", and the content in the database was correctly replaced	Pass

4.2.4 Order and Payment

TC-ORDER-001 calls OrdersService.add(), generates order flow and executes it completely.

TC-ORDER-002 executes ordersService.updateById(orders), successfully modifies order status.

TC-ORDER-003 hit ordersService.updateAccountById(), status updated to refunded.

Table 4 Order and Payment Test Result Table

Test Case ID	Test Step Description	Test Data	Expected Result	Actual Result	Pass/Fail
TC-ORDER-001	Select a travel route, click order, and submit	Product ID: 001 Number of people: 2	Generate an order with status to be paid	Hitting ordersMapper.insert() in OrdersService.add(), inventory verification passed, and the order was successfully generated	Pass
TC-ORDER-002	Simulate payment button click	Order number: O12345	The payment is successful and the status changes to Waiting for Shipment	Hit ordersService.updateById(), update the status field and return success	Pass
TC-ORDER-003	Click Refund after payment	Order status: Waiting for shipment	The status changes to Refunded	Hit ordersService.updateAccountById(), perform inventory rollback, and update the status to "Refunded"	Pass
TC-ORDER-004	Click Cancel before payment	Order status: To be paid	The status changes to Cancelled	ordersService.updateById() is called, and the order status is successfully updated to "Cancelled"	Pass

4.2.5 Feedback and Reply

TC-FEED-001 calls FeedbackService.add(), no exception is thrown, and Result.success() is returned

TC-FEEDBACK-002 hits FeedbackService.updateById(), and the backend reply content is saved and displayed in the frontend user feedback details.

TC-FEEDBACK-003 hits FeedbackService.deleteById(), the data is deleted from the database, the page feedback is "deleted successfully", and the table is updated.

Table 5 Feedback and Reply Test Result Table

Test Case ID	Test Step Description	Test Data	Expected Result	Actual Result	Pass/Fail
TC-FEEDBACK-001	Fill in the title and content and click the Submit button	Title: "Great service", Content: "Very helpful!"	"Operation successful" is displayed, and the feedback information appears in the feedback table	Feedback inserted successfully, visible in table, success message displayed	Pass
TC-FEEDBACK-002	The administrator opens the user feedback, enters the reply, and clicks Save	Response content: "Received and processed"	The reply is displayed in the corresponding feedback record	Feedback record updated, reply content saved and displayed appropriately	Pass
TC-FEEDBACK-00	Click the	Feedback	The	Feedback	Pass

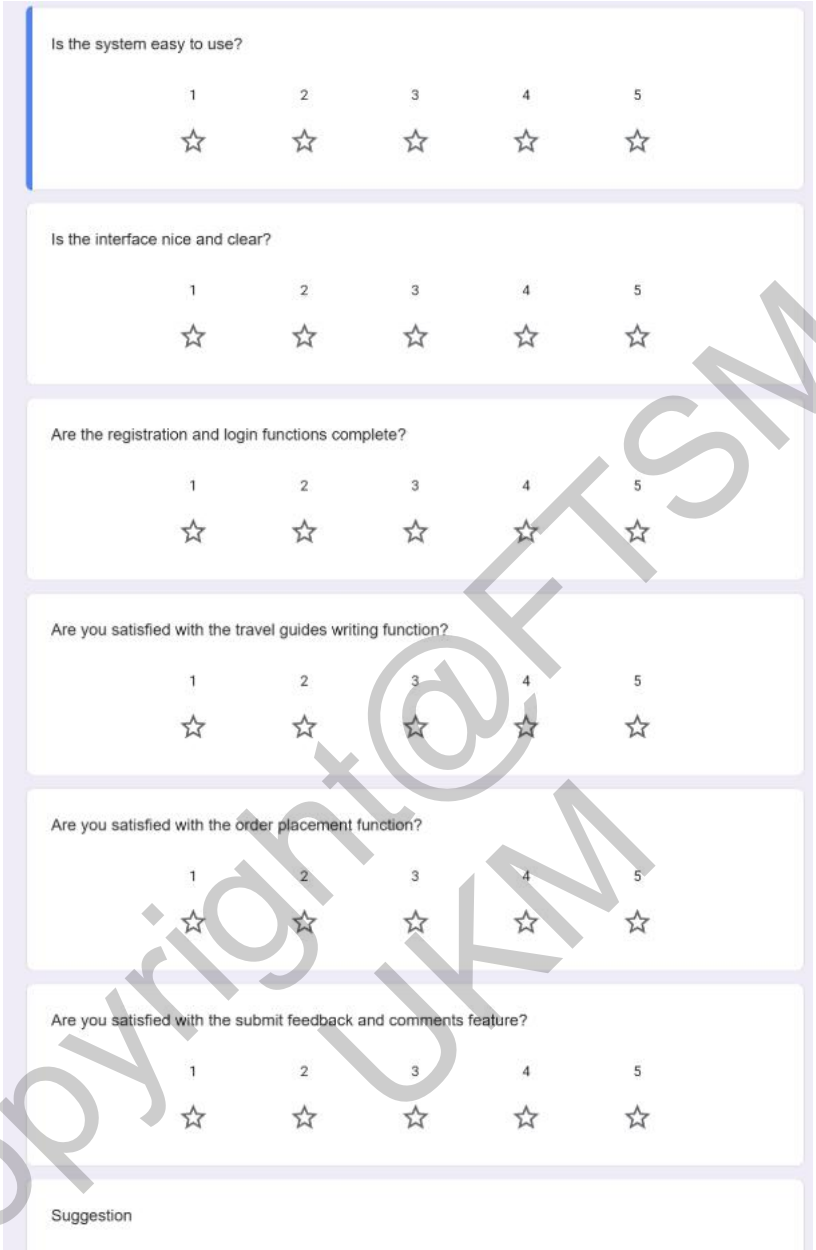
3	Delete button and confirm to delete the feedback	ID: 001	feedback item disappears from the table, and a message "Deleted successfully" appears	successfully removed from database, deletion message confirmed

User Acceptance Testing

To evaluate whether the system meets the actual needs and expectations of end users, we conducted User Acceptance Testing (UAT). This testing focused on user satisfaction with the interface design and user interaction with key features.

We invited 10 participants to conduct the test as typical users. Each user was assigned a set of predefined tasks. After completing the tasks, the user fills in the feedback form to rate the experience from 1 (poor) to 5 (excellent).

Figure 6 UAT form



4.3 RESULT

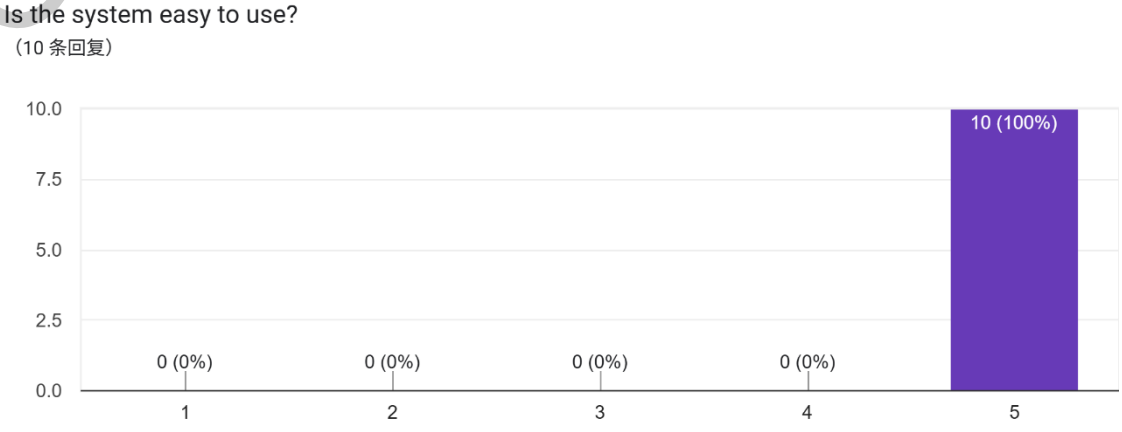
Table 5 Test Result Table

Module	Results	Conclusion
User Registration & Login	All test cases passed	Front-end validation and back-end logic work well

		together, can correctly handle registration exceptions and login verification, JWT interceptor functions normally
Travel Guides Writing	All tests passed, abnormal paths were blocked or handled	All tests passed, abnormal paths were blocked or handled
Order and Payment	Status transfer is correct, inventory synchronization is accurate, and all tests passed	The system can correctly handle order status changes, and the payment and cancellation processes are in line with expectations
Feedback	The test function is executable and the status is consistent	Administrator operations respond promptly and authority control is reasonable

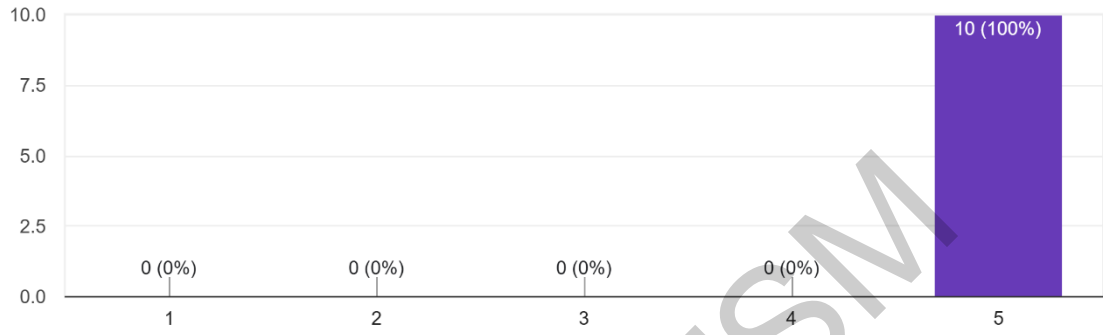
Final Verdict: Test Pass Rate: 100%

User Acceptance Testing Result:



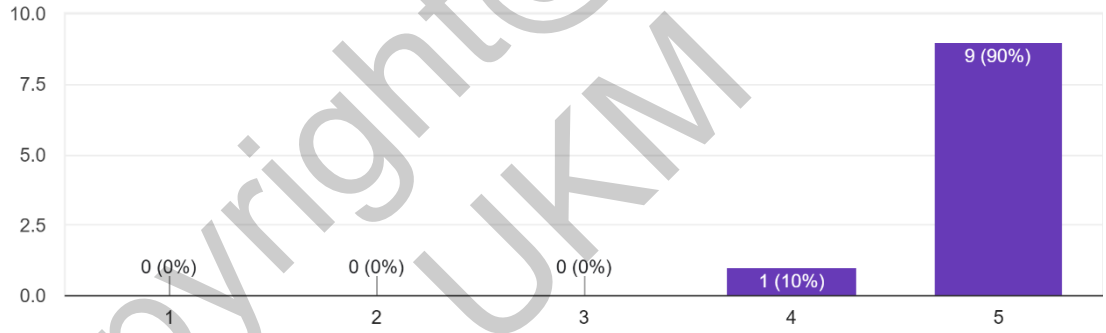
Is the interface nice and clear?

（10 条回复）



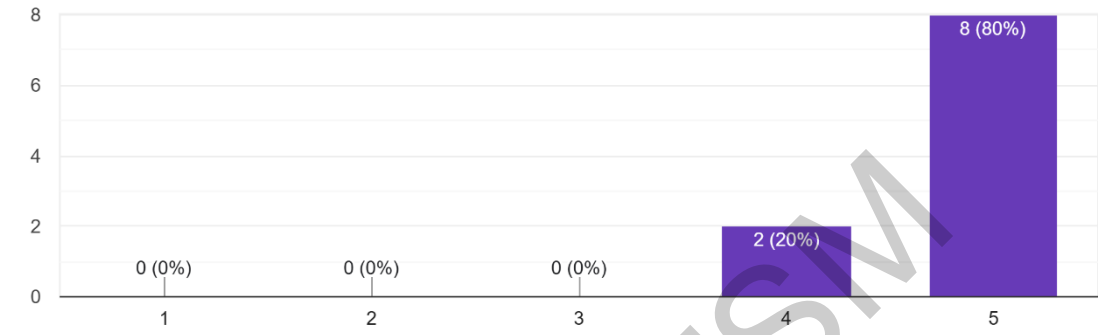
Are the registration and login functions complete?

（10 条回复）



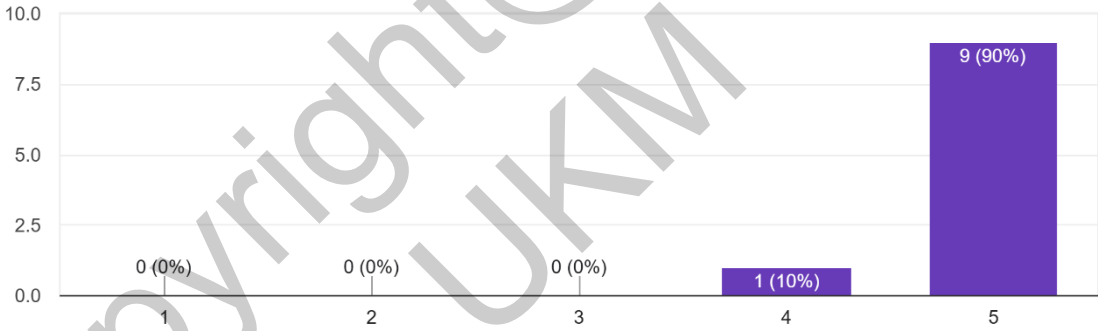
Are you satisfied with the travel guides writing function?

（10 条回复）



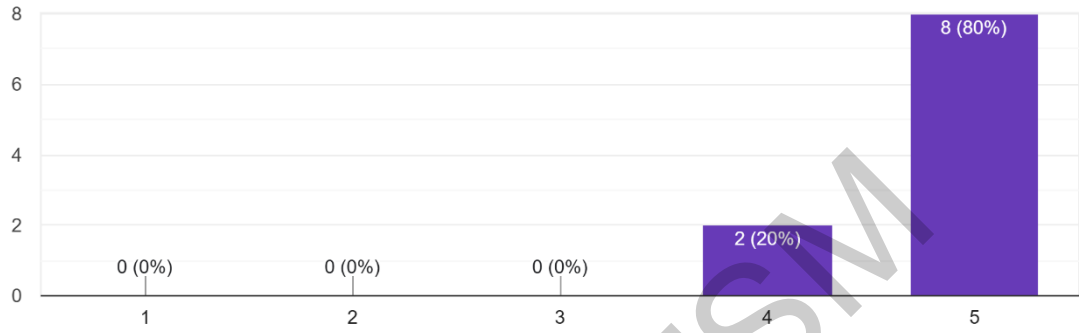
Are you satisfied with the order placement function?

（10 条回复）



Are you satisfied with the submit feedback and comments feature?

（10 条回复）



Suggestion

（7 条回复）

Feedback and comments cannot upload images

The number of people cannot be modified in the "To be paid" state

Users cannot view administrator response time

The column for writing travel guides is not easy to find

When setting a password, you can require it to contain uppercase and lowercase letters and numbers

If the travel guides is rejected, the administrator needs to explain the reason.

After the feedback question is answered by the administrator, the homepage can immediately indicate that it has been answered

Table 4.13 UAT Result Table

Survey Question	Number of 5 ratings (%)	Average Score
Is the system easy to use?	10 (100%)	5.0
Is the interface nice and clear?	10 (100%)	5.0
Are the registration and login	9 (90%)	4.9

functions complete?		
Are you satisfied with the travel guides writing function?	8 (80%)	4.8
Are you satisfied with the order placement function?	9 (90%)	4.9
Are you satisfied with the feedback submission and response feature?	8 (80%)	4.8
Average	9	4.9

These results demonstrate that the system meets users' expectations in terms of usability and function, with only minor improvements needed. The average scores for all questions were 4.9 out of 5, indicating a high level of user acceptance and satisfaction.

5.0 CONCLUSION

This project designed and developed a travel diary and service system. It is a web-based system that aims to help tourists provide a platform for mutual communication and make their travel easier. Its main functions include user registration and login, publishing travel guides,

placing and managing orders, and submitting feedback. The system uses Vue and Spring Boot for front-end and back-end separation development to ensure user and administrator interaction and secure data processing. The robustness of system operation and exception handling was verified through extensive testing such as white box testing, usability testing, and user acceptance testing.

6.0 REFERENCES

Şahin, S. (2012). *Intercultural Communication Efficiencies of Tourist Guides:*

Naif Almakayeel. (2023). *Relationship Modeling of Travel Website Quality toward Customer Satisfaction Influencing Purchase Intention*

ZHANG ZHIHANG (A197543)

Prof. Dr. Shahrul Azman Mohd Noah

Faculty of Information Technology & Science

National University of Malaysia