Bachelor of Computer Science

CSIT314 Software Development Methodologies

Project: Travel Platform

|  |  |
| --- | --- |
| Team Member | |
| Tam Hin Pak | 5976807 |
| Chan Lok Ching | 8563196 |
| Chu Kai Chit | 8711136 |

# Presentation Link: https://youtu.be/o49pxI-CYW8

# Table of Contents

[**Table of Contents 2**](#_heading=h.6mzob72f6nha)

[**Part I: Project Overview 3**](#_heading=h.5dxxugsaq50c)

[**Part II: Project Details 3**](#_heading=)

[**Part III: Methodologies 6**](#_heading=)

[**Part IV: Requirements Specification 8**](#_heading=)

[System Feature 1 : Login System 8](#_heading=h.2kqbmfa99h8j)

[System Feature 2 : E-commerce Features 9](#_heading=h.9isw2ouugdng)

[System Feature 3 : Blog Integration 10](#_heading=h.2bj22dn1yuk)

[**Part V: Time Management 11**](#_heading=)

[Gantt Chart 11](#_heading=h.bcbs0juuyfy)

[**Part VI: Software design 12**](#_heading=h.qo35phpjn2nb)

[BCE model 12](#_heading=h.x9wnsaly0fk5)

[Use Case Diagram 15](#_heading=h.suidzj57c29v)

[Activity Diagram 17](#_heading=h.oacsj4qn4x2f)

[State Machine Diagram 18](#_heading=h.hg2vjbqcha99)

[BCE Sequence Diagram 19](#_heading=h.wolesethtctx)

[**Part VII: Test plan 23**](#_heading=h.5ff8kig0vtvq)

[Module 1: Account Related 23](#_heading=h.3xxe9b3z22ib)

[Module 2: E-commerce Related 24](#_heading=h.y80jam1wqb2t)

[Module 3: Comment Page and Reply System 25](#_heading=h.78s8k81tdv5s)

[**Part VIII: Description for gathering ideas-tool 26**](#_heading=h.xmbprsfw9bbf)

[**Part IX: Stakeholders Analysis 26**](#_heading=h.42spqi514y15)

[Internal Stakeholders 26](#_heading=h.vefd4vigta0h)

[External Stakeholders 26](#_heading=h.ws6pbsa779k7)

[**Part X: Roles and responsibilities 27**](#_heading=h.exkvxzt58q4o)

[**Appendix 28**](#_heading=h.ipwm3lw02q2c)

[Meeting minutes 28](#_heading=h.re5uzbx627n3)

[Source Code 30](#_heading=h.lp1hmhhlweiu)

Project Charter

# Part I: Project Overview

|  |  |
| --- | --- |
| **Student Name & ID** | **Tam Hin Pak 5976807**  **Chan Lok Ching 8563196**  **Chu Kai Chit 8711136** |
| **Project Name** | **Developing a travel platform for sharing different information of travel** |
| **Project Charter Author** | **Mr.Tam Hin Pak\***  **Mr.Chan Lok Ching**  **Mr.Chu Kai Chit**  **\* project leader** |
| **Creation Date** | **27 Dec, 2023** |
| **Last Revision Date** | **20 Mar, 2024** |
| **Project Charter Status**  **(Pending/Approve/Reject)** | **Pending** |
| **Date of Project Approval** | **20 Mar 2024 (exp)** |
| **Proposed Project Start & End Date** | **27/Dec/2023**  **20/Mar/2024** |

# Part II: Project Details

|  |  |
| --- | --- |
| **Project Description** | **TravelPlan is a platform that develops a cutting-edge travel platform, offering a comprehensive range of features and services to empower travelers worldwide. The aim of this project is to provide the information to share with different users. Additionally, the users can provide some service with their blog information.** |
| **Project Purpose** | * **Provide a user-friendly interface that caters to the needs of travelers worldwide** * **Provide a wide array of functionalities and services to cater to the diverse needs of travelers** * **Design the platform to cater to travelers from diverse cultural backgrounds, languages, and preferences** * **Enable travelers to access valuable travel information, tips, and recommendations shared by other users** |
| **Project Goals & outcomes** | * **Goal:**   + **To provide an interactive and attractive platform that engages different travelers, we encourage them to share their travel stories on our platform. These stories serve to attract and connect with various users who may have a similar interest or need, such as reading Ben’s travel plan for Tokyo. In addition to sharing their experiences, these travel plans include suggested routes, different travel approaches, notable attractions, and recommendations for local food.**   + **The aim of our platform is to offer travel information provided by fellow travelers and help solve the questions that arise before embarking on a journey.**   + **To be completed in 3 months.** * **Outcome:** * **Users will have access to a user-friendly interface that simplifies the process of searching, comparing, and booking flights, hotels, and other travel services. This saves time and effort in planning their trips.** * **The platform will provide personalized travel recommendations based on users’ preferences, travel history, and feedback. This feature helps users discover new destinations, and experiences that align with their interests.** * **Users will have access to curated destination guides that offer comprehensive information, local insights, and recommendations for a wide range of locations. These guides help users make informed decisions about their travel destinations and explore the best attractions and activities.** * **The platform will prioritize user data privacy and transactional security. Users can book their travel services with confidence, knowing that their personal and financial information is protected.** |
| **Project Scope** | * **User-friendly interface for searching, comparing, and booking flights, hotels, and other travel services.** * **Personalized recommendations and itineraries based on user preferences, travel history, and feedback.** * **Curated destination guides with comprehensive information, local insights, and recommendations.** * **Real-time reviews and ratings from a community of travelers.** * **Social features for users to connect, share experiences, and engage in discussions.** * **Secure booking and payment systems to ensure user data privacy and transactional security.** * **Mobile-friendly platform for seamless access from desktop and mobile devices.** |
| **Project Deliverables** | **Functional deliverables**   * **Login/ Register**    + **Develop a secure environment**   + **Allow user to logout and login to other account**   + **Edit personal profile information** * **Maps Feature**   + **Displaying maps and adding makers for travel destinations**   + **Zoom-in/ Zoom-out for interactive** * **Commenting platform**   + **Post comments**   + **Reply existing comments** * **Filtering and Categorization**   + **Categorize blogs and products**   + **Search to find relevant contents** * **Package Creation and Showcase**   + **Create, edit the package and publish** * **E-commerce Functionality**   + **Set up for buying products**   + **Integrate a shopping cart**   + **Payment Gateway** |
| **Benefits for Customers** | * **Brand Building** * **Increase Customer Engagement** |
| **Benefits for Users** | * **Easier to find desired tour packages or blog posts** * **Personalized Recommendations** |
| **Stakeholders** | **Internal:**  **Project Sponsors**  **Project Manager**  **Business Analyst**  **Developers**  **Programmers**  **Customers**  **External:**  **Traveler**  **Website platform owner**  **Membership Members** |
| **Assumptions** | **User Internet Connectivity:Assume that the majority of target users will have reliable internet connectivity to access and use the TravelPlan platform.**  **User Language Proficiency: Assume that a significant portion of target users have a basic understanding of the English language.** |
| **Project Team** | **Project Owner**  **Scrum Master**  **Team(Business Analyst, Developers, Programmers)** |
| **Budget Requirements** | **Website Design - HK$ 40,000 / 2 months**  **Payment cooperation - HK$3, 000/ each month** |

# Part III: Methodologies

In our project, we will be adopting the Agile methodology, specifically during the development and deployment stages of the TravelPlan platform. The development process will be divided into several iterative segments known as sprints, each with its own set of defined objectives. These segments encompass planning, design, development, testing, deployment, and review.

For the design of our platform, we will utilize HTML, CSS, and JavaScript.

|  |  |
| --- | --- |
| **Sprint 1** | |
| Plan | * Define the project goals, scope, and target audience * Identify the core features and functionalities required for the initial version of the platform |
| Design | * Design the user interface (UI) with a focus on usability and intuitive navigation |
| Develop | * Implement user registration and authentication functionality * Set up a basic user profile management system * Create a blog creation interface with text formatting capabilities |
| Test | * Conduct unit testing to verify the functionality of implemented features * Collect user feedback on the initial features to identify any usability issues or bugs |
| Deploy | * Set up the necessary infrastructure and configure databases * Conduct final testing in the staging environment |
| Review | * Evaluate the platform's performance, functionality, and user experience in the staging environment * Collect feedback and make the necessary adjustments |

|  |  |
| --- | --- |
| **Sprint 2** | |
| Plan | * Review user feedback and prioritize feature enhancements and bug fixes |
| Design | * Improve the visual design and layout based on user feedback and design best practices * Design the commenting system to encourage user engagement and interaction |
| Develop | * Implement personalized recommendation functionality based on user preferences * Develop an interactive commenting system, allowing users to post comments and replies * Integrate filtering and search functionality to improve content discovery |
| Test | * Conduct user acceptance testing to ensure the changes meet user expectations * Address any identified issues or bugs through bug fixing and iterative improvements |
| Deploy | * Perform the necessary infrastructure and database updates |
| Review | * Evaluate the performance, functionality, and user experience of the updated platform in the staging environment * Collect feedback and incorporate it into further improvements |

|  |  |
| --- | --- |
| **Sprint 3** | |
| Plan | * Gather user feedback on the previous iterations and prioritize further improvements * Explore integration options for an e-commerce system |
| Design | * Design the e-commerce functionality, including product listings and inventory management |
| Develop | * Implement e-commerce integration including Shopping cart and the ability to edit quantity, allowing users to offer services and sell products * Enhance the search and filtering capabilities for improved content discovery |
| Test | * Conduct comprehensive testing of all implemented features and functionalities |
| Deploy | * Perform the necessary infrastructure and database updates |
| Review | * Conduct regular reviews and monitoring to address any emerging issues |

|  |  |
| --- | --- |
| **Sprint 4** | |
| Plan | * Gather user feedback on the previous iterations and prioritize further improvements * Evaluate integration options for payment gateway |
| Design | * Design the payment gateway functionality, including payment process and payment record * Ensure that the design aligns with industry standards and best practices for secure and seamless payment experiences |
| Develop | * Implement the integration of the selected payment gateway, enabling users to securely complete their transactions during the checkout process. * Develop the necessary logic to automatically calculate the total fee based on the products selected by the user |
| Test | * Conduct stress test of all implemented features and functionalities |
| Deploy | * Perform final testing and ensure all components are functioning correctly * Prepare the final version of the platform for deployment to the production environment |
| Review | * Continuously engage in bug fixing and refinement based on user feedback and testing results * Conduct thorough quality assurance checks to ensure that the platform meets the highest standards of reliability, security, and user satisfaction |

In summary, the iterative development approach has significantly enhanced the TravelPlan platform by incorporating user feedback and continuously refining its features. This process has led to the implementation of valuable functionalities, swift resolution of issues, and a high level of user satisfaction. As a result, TravelPlan now offers a user-friendly platform that effectively meets the needs of its audience. Ongoing reviews and further improvements will continue to bolster TravelPlan's success in the travel industry.

# Part IV: Requirements Specification

In this document, we will outline the functional requirements for the system, organized based on the following system features: login system, e-commerce integration, and blog integration. The purpose of this specification is to provide a clear understanding of the system's functionality and guide the development process.

## System Feature 1 : Login System

**Description and Priority**

The login system serves as a fundamental component of the platform, allowing users to securely access their accounts and utilize personalized features. It is of high priority to ensure a seamless and secure user authentication process.

**Stimulus/Response Sequences**

* Stimulus: The user accesses the platform's login page.
* Response: The system presents a login form for the user to enter their credentials.
* Stimulus: The user enters their username and password and submits the login form.
* Response: The system verifies the user's credentials in local storage and grants access to the user's account if they are valid.
* Stimulus: The user requests to reset their password.
* Response: The system provides password reset functionality, guiding the user through a secure process to reset their password.

**Functional Requirements**

|  |  |
| --- | --- |
| FREQ-1:User-Friendly Login Page | The system shall provide a user-friendly login page with clear instructions and intuitive input fields for users to enter their credentials. |
| FREQ-2:Secure Storage of User Credentials | The system shall securely store user credentials, employing encryption and hashing algorithms to protect sensitive information such as passwords. |
| FREQ-3:Validation of User Credentials | It should verify the accuracy of the provided username and password combination and grant access to the user's account if the credentials are valid. Invalid credentials should result in an appropriate error message. |
| FREQ-4:Password Reset Functionality | The system shall allow users to securely reset their passwords in case they forget or need to change them. |

## System Feature 2: E-commerce Features

**Description and Priority**

The e-commerce features enable users to browse and purchase tour packages seamlessly through the platform. Authorized users can add new packages and edit existing packages as well. These features are of high priority as they directly contribute to the platform's revenue generation and user satisfaction.

**Stimulus/Response Sequences**

* Stimulus: The users add a tour package to their shopping cart.
* Response: The system updates the shopping cart, displaying the added package and calculating the total cost.
* Stimulus: The user edits the quantity of the tour package.
* Response: The system modifies the amount and recalculates the total cost.
* Stimulus: The user proceeds to checkout.
* Response: The system initiates the checkout process, guiding the user through the necessary steps to complete the purchase.
* Stimulus: The user provides payment information and confirms the transaction.
* Response: The system securely processes the payment, verifies its success, generates an order confirmation for the user, and stores the payment record for future reference.

**Functional Requirements**

|  |  |
| --- | --- |
| FREQ-1:Shopping Cart Management | The system shall allow users to add tour packages to a shopping cart, view the contents of the cart, update quantities, and remove packages if desired. |
| FREQ-2:Checkout Process | The system shall guide users through the checkout process, collecting necessary information, such as billing details, and providing a summary of the order. |
| FREQ-3:Order Confirmation | The system shall generate an order confirmation for users upon successful completion of the purchase. |
| FREQ-4:Order History and Tracking | The system shall maintain an order transaction history for each user, allowing them to view past orders and track the status of current orders. |
| FREQ-5: Create and Modify Package | The system shall create a set of forms and allow authorized users to enter their packages on the platform. |

## System Feature 3: Blog Integration

**Description and Priority**

The blog integration feature enhances the platform by providing users with informative and engaging blog content related to travel and tour experiences. It is of medium priority, as it contributes to user engagement and provides valuable information but may not directly impact revenue generation.

**Stimulus/Response Sequences**

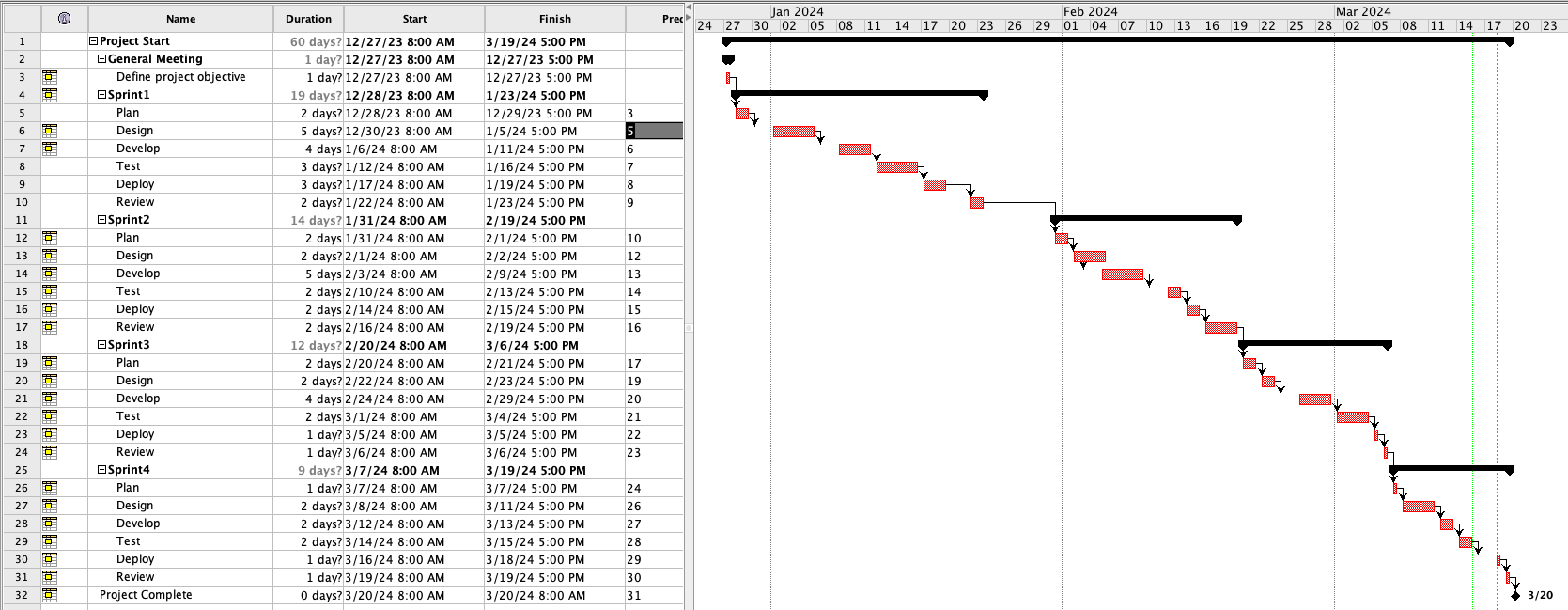
* Stimulus: The user selects a blog post to read from the corresponding package.
* Response: The system displays the selected blog post, including the title, content, and any associated images or media.
* Stimulus: The user interacts with the blog post by leaving comments and zooming in on the map.
* Response: The system enables logged-in users to engage with the blog post.

**Functional Requirements**

|  |  |
| --- | --- |
| FREQ-1:Commenting Functionality | The system shall provide blog comment posts and replies, allowing users to engage in discussions. |
| FREQ-2: Map Marker | The system shall provide a map interface for displaying markers related to tour packages or points of interest. |
| FREQ-3: Related Posts and Recommendations | The system shall analyze user preferences, browsing history, and engagement patterns to identify related blog posts. |

# Part V: Time Management

## Gantt Chart



# Part VI: Software design

## BCE model

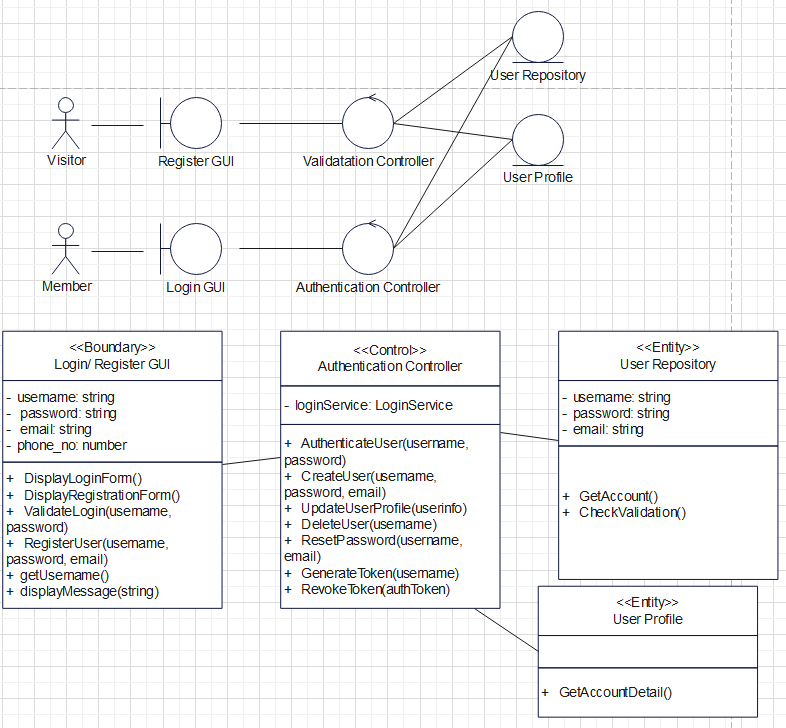


Figure 1: Login/Register Interface B-C-E Model

**Boundary:**

The boundary represents the user interface of the application responsible for handling user authentication and registration. It provides a visual interface for users to interact with the system, including displaying login and registration forms. It also includes methods for validating login credentials and registering new users.

**Control:**

The authentication controller is responsible for handling operations related to user authentication and user management. It receives input from the boundary and processes the requests accordingly. The controller includes methods for authenticating users, creating new user accounts, updating user profiles, deleting user accounts, resetting passwords, and generating and revoking authentication tokens.

**Entity:**

represents the storage and retrieval of user account data. It is responsible for managing user account information such as usernames, passwords, and email addresses. The user repository and profile ensure the persistence and integrity of user account data.

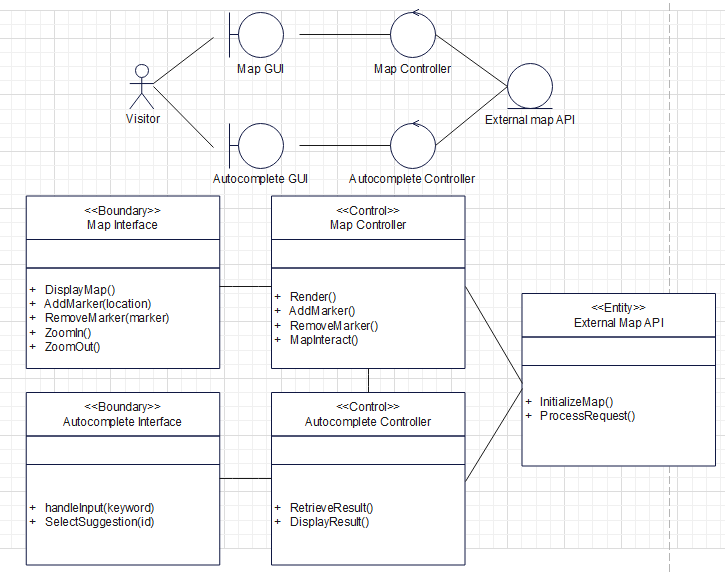


Figure 2: Map Interface B-C-E Model

**Boundary:**

This represents the interface that displays maps and allows users to interact with them. It includes displaying travel destinations and adding markers to the map. The boundary includes methods for rendering the map, adding markers to specific locations, and autocomplete results based on user input.

**Control:**

This handles the operations related to the map functionality. It receives input from the map and processes the requests accordingly. It includes methods for rendering the map, adding markers to specific locations, and capturing and responding to user interactions with the map, such as dragging markers or zooming in/out, and providing suggestions to the input field for display.

**Entity:**

This represents the external service provided by Leaflet for map-related functionality. It provides the necessary tools and resources to render maps, add markers, and perform autocomplete searches.

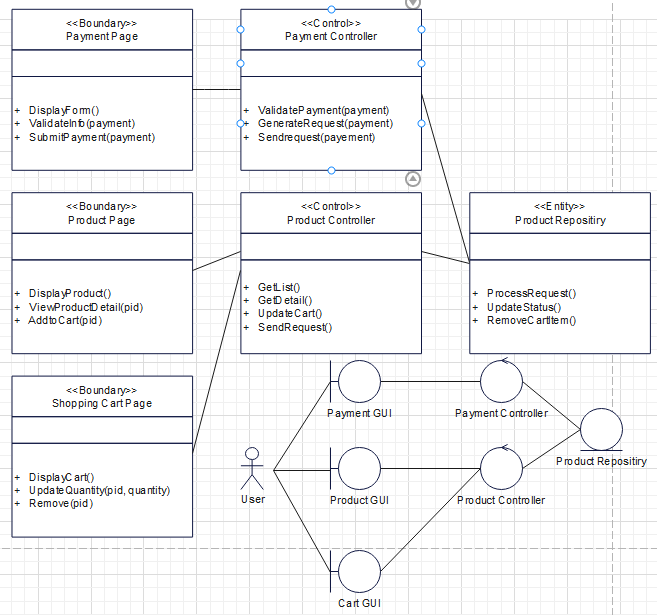


Figure 3: Product and Shopping Cart Interface B-C-E Model

**Boundary:**

The Payment interface handles collecting payment information, while the Product interface displays available products and allows users to view details and add items to the shopping cart. The Cart interface provides a summary of items in the cart and allows users to modify its contents.

**Control:**

The Payment Controller is responsible for validating payment information and processing transactions, and the Product Controller manages product-related operations such as retrieving listings, fetching details, and adding items to the cart. These components collaborate to create a seamless and efficient e-commerce experience.

**Entity:**

The entity in e-commerce is the Product Repository. It handles data operations for product management, such as retrieving and updating product information from the underlying data storage.

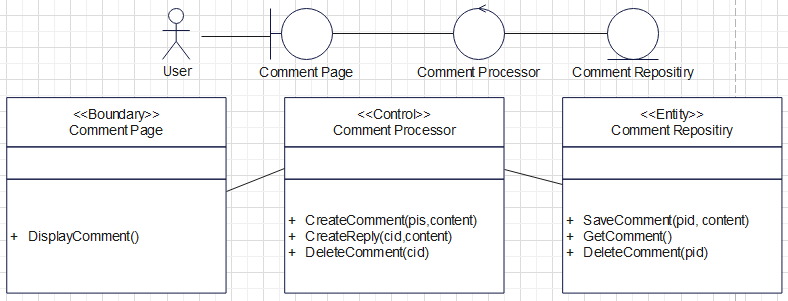


Figure 4: Comment Interface B-C-E Model

**Boundary:**

This represents where users can view and leave comments. It provides functionalities for displaying post comments and reply comments, allowing users to read, write, and delete comments.

**Control:**

It manages the operations related to the commenting feature. It receives input from the Comment Boundary and is responsible for processing user actions such as posting comments, replying to comments, and deleting comments.

**Entity:**

This represents the comment data structure and its associated properties. It encapsulates handling comments, including storing, retrieving, and deleting comments. The comment entity interacts with the underlying data storage, such as a database, to persist and retrieve comment data.

## Use Case Diagram

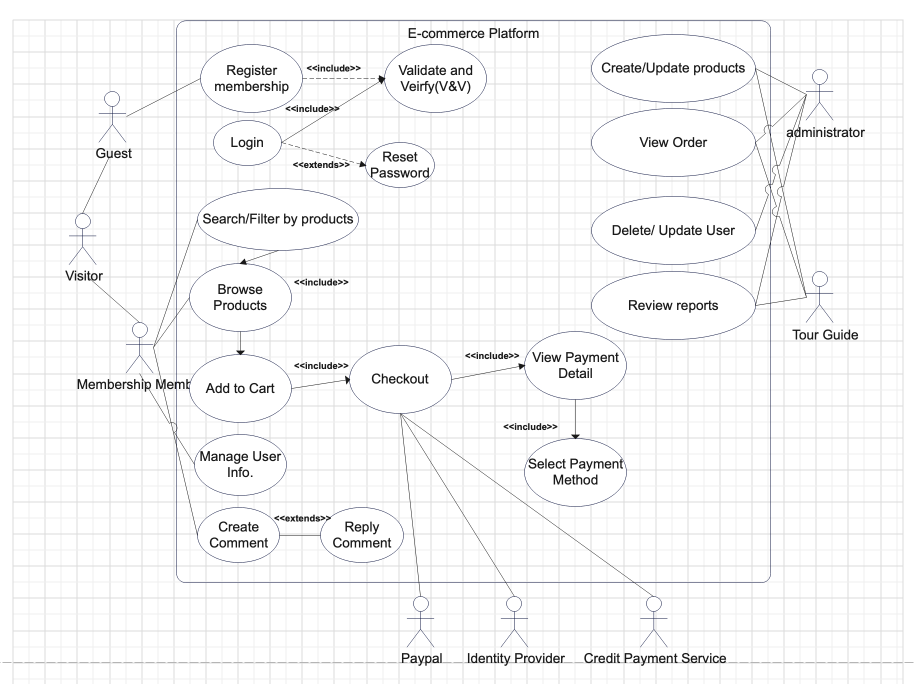


Figure 5: Use Case Diagram for the platform

Use Case 1

|  |  |
| --- | --- |
| Name: Membership | ID: 1 |
| Description:  The guest registers the membership by providing the necessary information and, after that, verifying and validating their account. | |
| Precondition:  The guest is a visitor on the e-commerce platform. | |
| Normal Flow:   1. The guest selects the option to register for the membership programme. 2. The e-commerce platform prompts the guest to provide the required information. 3. The guest submits the information. 4. The e-commerce platform verifies the information and creates a Membership Member account. | |

Use Case 2

|  |  |
| --- | --- |
| Name: Shopping Cart | ID: 2 |
| Description:  The membership member or guest browses and adds products to the shopping cart. | |
| Normal Flow:   1. The user selects the option to browse products or trips. 2. The e-commerce platform displays available products or trips. 3. The member can use filters or the feature of searching to refine search results. 4. The member selects desired products or trips and adds them to the shopping cart. 5. The member can repeat steps 2-4 until all desired products or trips are added. | |

Use Case 3

|  |  |
| --- | --- |
| Name: Process Payment | ID: 3 |
| Description:  The member proceeds to process payment for the items in the shopping cart. | |
| Precondition:  The member has added some products or trips to the shopping cart. | |
| Normal Flow:   1. The user selects the option to proceed to checkout. 2. The platform prompts the user to log in to their account. 3. If a user forgets the password, they can reset it by receiving a verification code through email. 4. The member logs in to their account. 5. The platform verifies the login credentials. 6. The member views the payment details and selects a payment method such as a credit card or debit card. 7. The member completes the payment process. | |

Use Case 4

|  |  |
| --- | --- |
| Name: Commenting System | ID: 4 |
| Description:  The membership member can create comments and reply to existing comments | |
| Precondition:  The membership member is logged into their account | |
| Normal Flow:   1. The membership member selects the option to create a comment or reply to an existing comment. 2. The platform prompts the membership member to enter the comment content. 3. The membership member submits the comment or reply. 4. The platform adds the comment or reply to the system. | |

## Activity Diagram

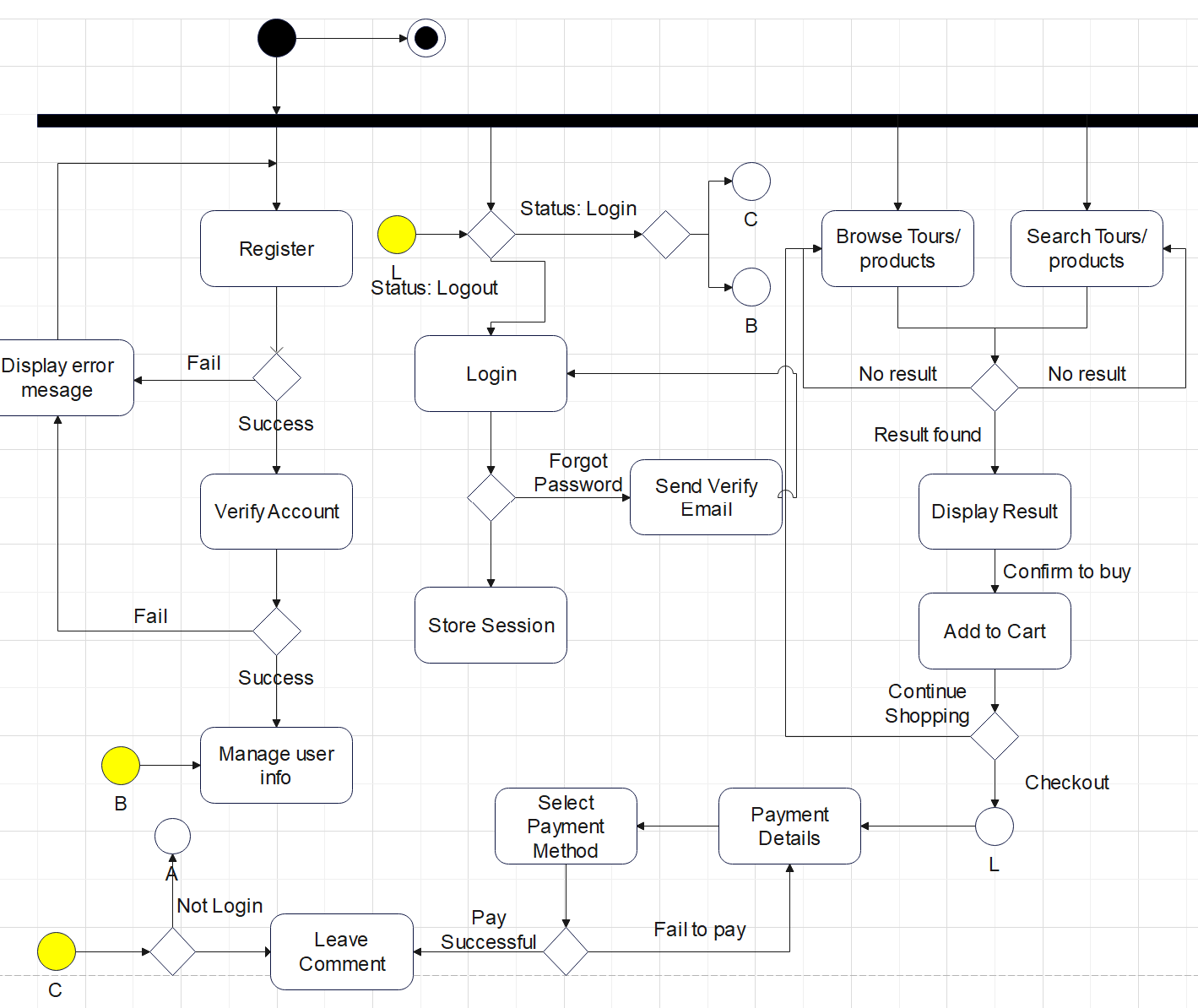


Figure 6: Activity Diagram for the whole platform

The activity diagram depicts three different states. When the user chooses to "register," they provide the necessary information, including a username, password, email, and phone number. Once the registration is successful, it triggers a transition to verify the account. In the event that the registration process encounters an error or the provided information is invalid, an error message is displayed. Upon successful verification of the user's account, it switches to the profile page, allowing the user to view or manage their user information.

First, when the user chooses to "browse tours and products," the system displays the details of the products. If the desired result is not found, the process goes back to the "browse tours and products" stage. Simultaneously, when the user chooses to "Search Tours/Products," the system displays the search results. If no results are found, the process also goes back to the "Search Tours/Products" stage.

Second, when the user confirms their intention to buy tours or products and adds them to the shopping cart, they can proceed to the checkout. If the user wants to continue shopping, they can go back to the "Browse Tours/Products" process.

Third, before the shopping cart switches to the checkout process, the system asks the user to log in to their account. If the user has not logged in, they need to do so before proceeding. In the checkout process, the system provides different payment details for the user to select their preferred payment method. If the payment is successful, the user can leave a comment or log out of their account. If the payment fails, the system leads the user back to the payment details process.

State Machine Diagram

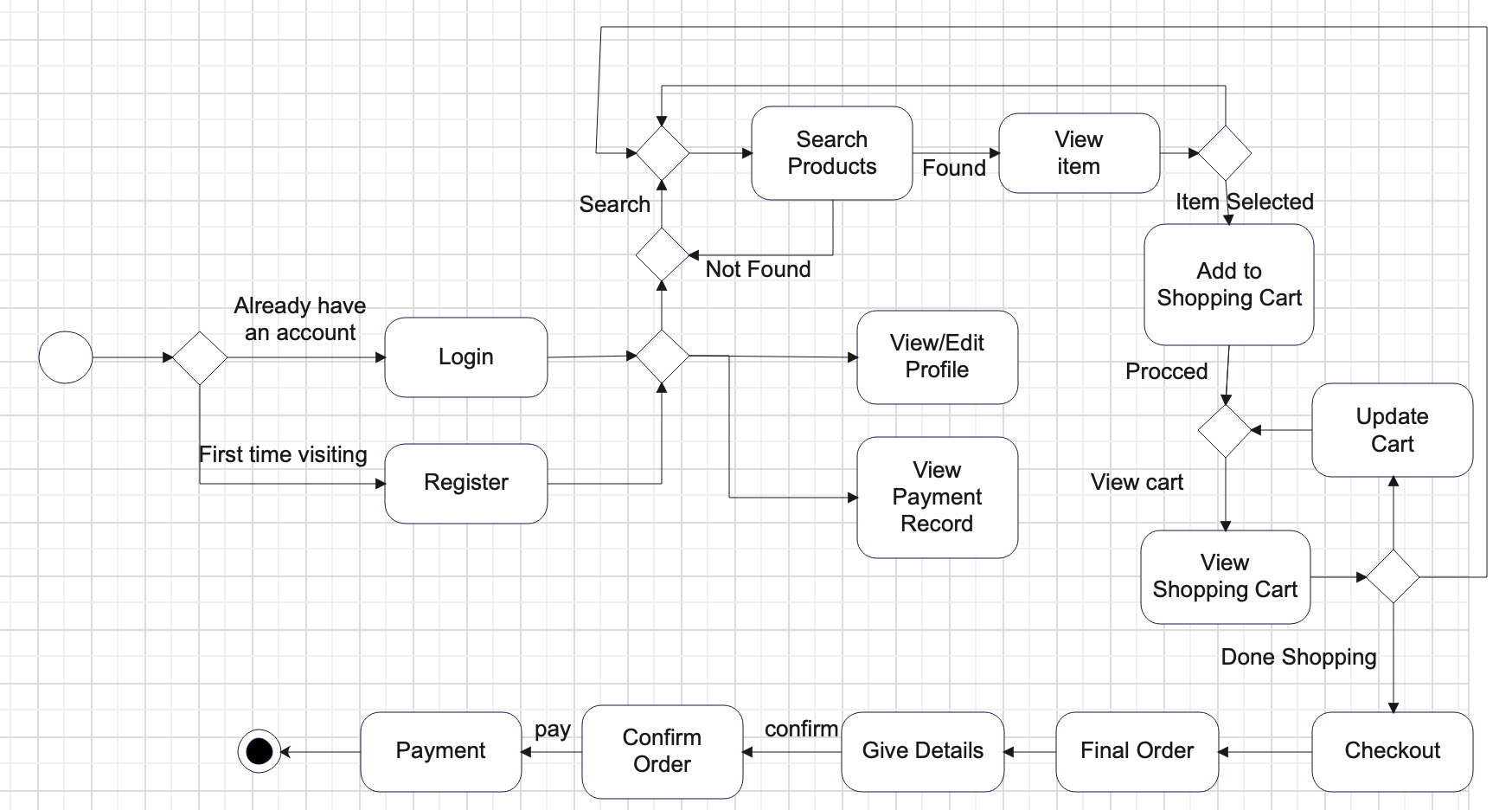


Figure 7: State Machine Diagram for the platform

The initial state marks the beginning of the process. When the user wants to access the system, they have the option to either "Login" or "Register." If the user selects "Login," they enter their credentials, and if the login is successful, it triggers a transition to the next process. If the login fails, the process concludes in the "Login Failed" state.

On the other hand, if the user chooses to "register," they provide the necessary information, including a username, password, email, and phone number. Once the registration is complete, it triggers a transition to the next process. In the event that the registration process encounters an error or the provided information is invalid, the process concludes in the "Registration Failed" state.

When the user selects "Search Product," they enter the relevant product keywords. If the product is found based on the search keywords, it triggers a transition to the "View Item" process. If the product is not found, the process loops back to the "Search Product" state.

When the user selects a product to add to the shopping cart, it triggers a transition to the "View Shopping Cart" process. If the user wishes to search for another product, the process loops back to the "Search Product" state.

Upon viewing the shopping cart details, the user can proceed to the "checkout" process. In the checkout process, the system provides order details. If the user is satisfied with the final order, it triggers a transition to the "Payment" process. If the payment is successful, the process concludes. Otherwise, the process concludes in the "payment" state.

## BCE Sequence Diagram

Login:

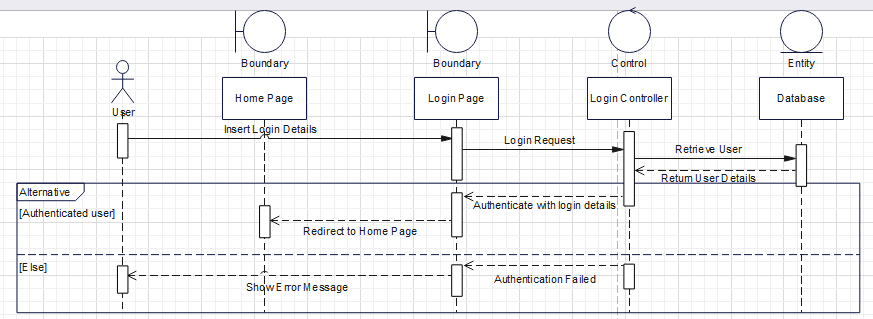


Figure 8: Sequence Diagram for login feature

When the user enters their login details on the login page, the page sends a request to the login controller. The login controller retrieves the user's information from the database and returns the user's details. Using these details, the login controller authenticates the user with the login page and redirects them to the home page. In the event that authentication fails at the login controller, the login page displays an error message to notify the user.

Register:

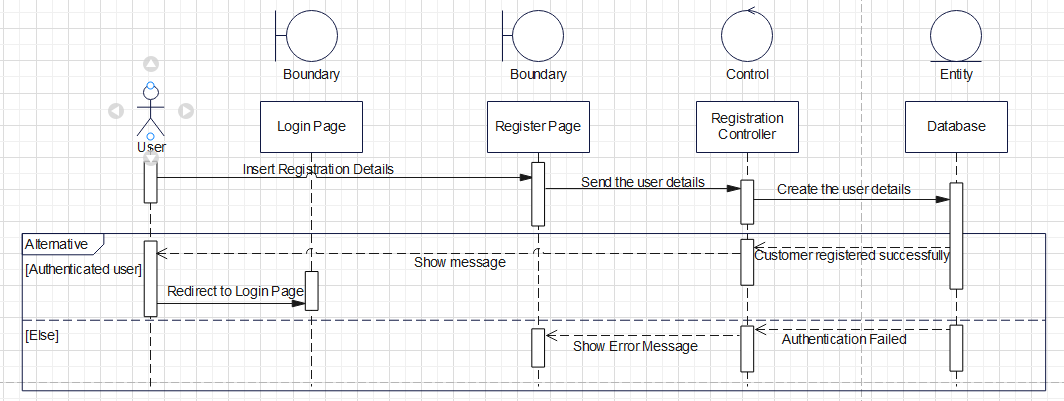


Figure 9: Sequence Diagram for register feature

When the user enters their registration details on the register page, the page sends the user details, such as username and password, to the registration controller. The registration controller then forwards the user details to the database, where the user details are created and stored. If the user details are successfully created in the database, the database displays a success message to the user and redirects them to the login page. However, if the database encounters an error during the registration process, it will display an error message to the user.

Browse product to Shopping Cart:

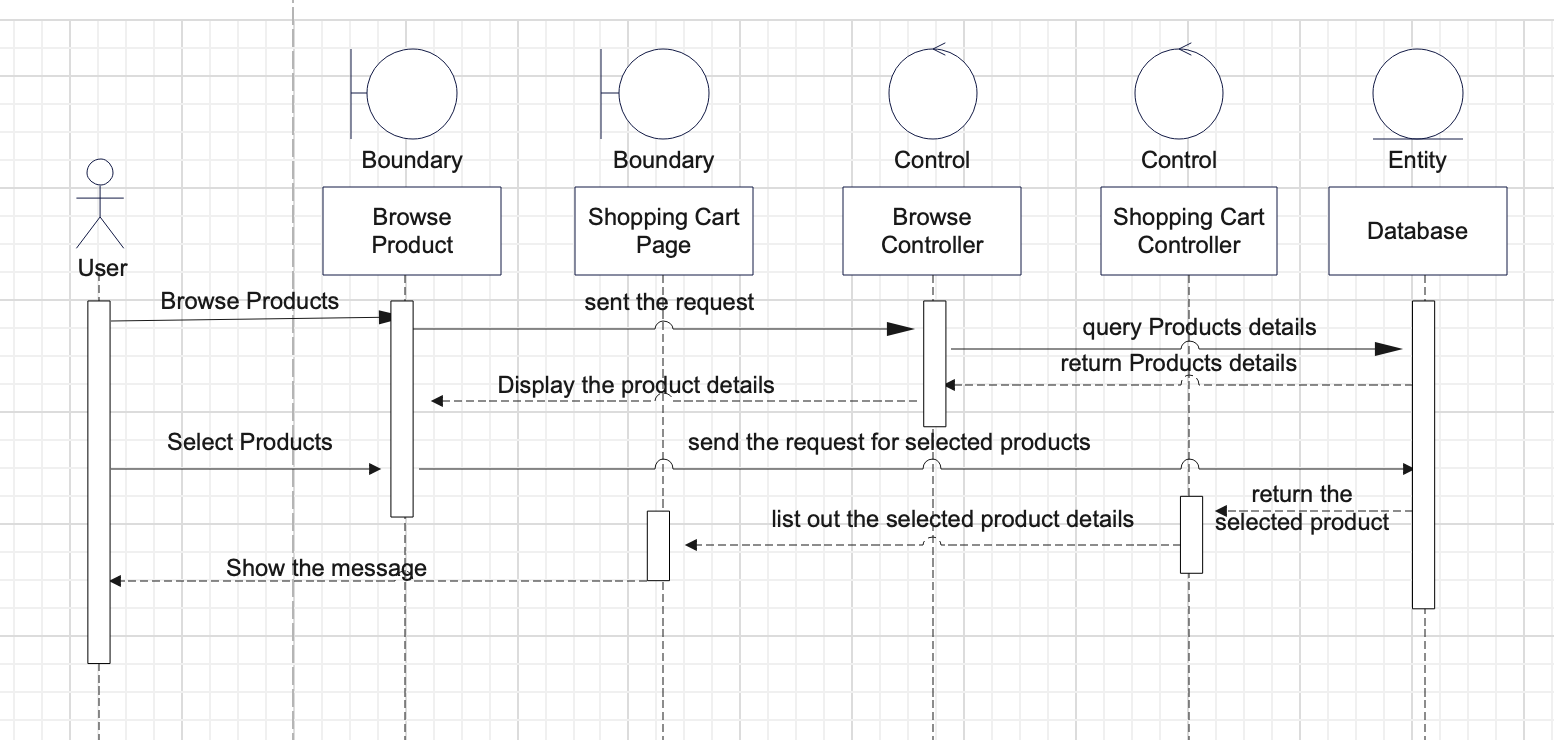


Figure 10: Sequence Diagram for e-commerce feature

First, the user accesses the product page to browse available products. The page sends a request to the "Browse Product" controller. The "Browse Product" controller queries the database for details of all products, retrieves the information, and returns it to the controller. Once the "Browse Product" controller receives the product details from the database, it displays them to the user.

Next, the user selects the desired products on the browse product page. The page sends a request for the selected products to the database. The database processes the request and returns the selected product information to the "Shopping Cart" controller. The "Shopping Cart" controller adds the details of the selected products and displays them on the shopping cart page. Upon successfully adding all the products, the "Shopping Cart" controller shows a success message to the user.

Shopping Cart to Checkout:

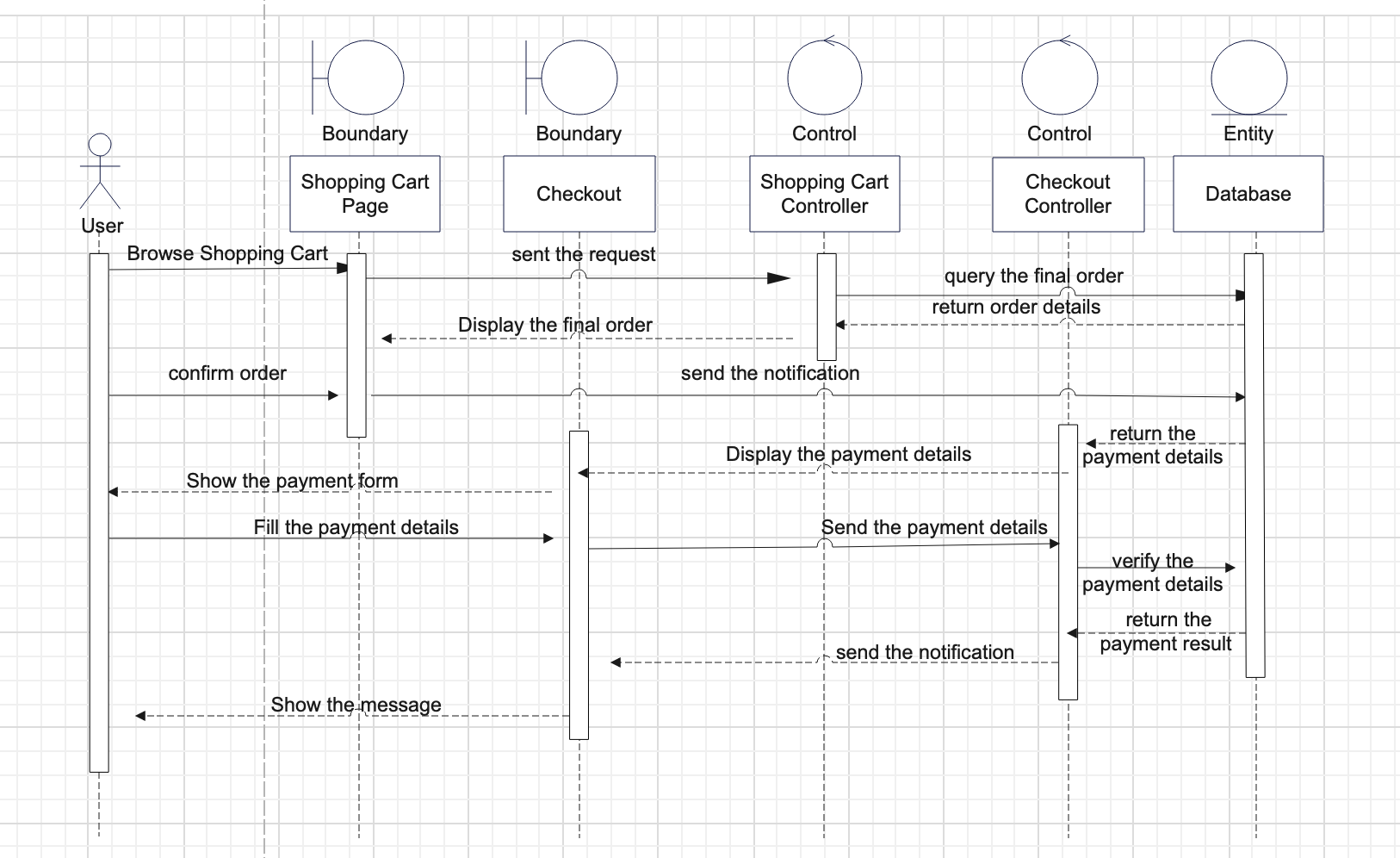


Figure 12: Sequence Diagram for the process from shopping cart to checkout

The user accesses the shopping cart page to review their items. Upon accessing the page, a request is sent to the "Shopping Cart" controller, which queries the database for the final order. The database returns the order details to the "Shopping Cart" controller, which automatically displays the final order to the user.

On the shopping cart page, the user confirms the order and sends a notification to the database. The database, in turn, returns the payment details to the controller. The checkout page then displays the payment details and presents the payment form to the user. The user fills in the payment details on the checkout page, which sends the payment details to the checkout controller for verification against the database.

Upon receiving the payment details, the database verifies their correctness and returns the payment result to the checkout controller. The checkout controller sends a notification to the checkout page, which displays a message to the user.

## 

# Part VII: Test plan

## Module 1: Account Related

|  |  |
| --- | --- |
| ID: TC01 | Test Case: Verify user login details |
| Purpose: To ensure that the user login functionality is working correctly and validating user credentials. | |
| Expected Results:   * The user entered a valid username and password and should be successfully logged in. * The user enters an invalid username or password, and an error message should be prompted. * The user leaves the username or password field blank, and an error message should be prompted. | |
| Actual Results:   * The user was successfully logged in and redirected to the home page for further exploration. * The system displayed the error message “Invalid username or password.”. * The system displayed the error message “Please input all fields.”. | |

|  |  |
| --- | --- |
| ID: TC02 | Test Case: Validate if inputs in register page are correct format |
| Purpose: To ensure that the system properly validates input fields for the correct format. | |
| Expected Results:   * The user enters all valid input formats, including the correct email format and phone number, which are 8-digit numbers, and redirects to the corresponding page. * The user should enter an invalid email address format, such as a missing ‘@’ symbol, and an error message should be prompted. * The user enters an invalid phone number format, and an error message should be prompted. | |
| Actual Results:   * The user successfully registers an account and redirects to the login page to login again. * The system displayed the error message “Invalid email address format.”. * The system displayed the error message “Invalid phone number format.”. | |

|  |  |
| --- | --- |
| ID: TC03 | Test Case: Verify profile page functionalities |
| Purpose: To ensure that the user's profile page is displayed correctly with accurate information. | |
| Expected Results:   * When a user has logged in, their username, email, and basic information should be displayed accurately for modifications. * When a user clicks on the submit button with no modifications, the system should prevent updating information. * When a user clicks on the submit button with modifications, the system should verify and update the information. | |
| Actual Results:   * The input fields automatically fill in relevant data for modifications. * The system displayed the error message “No changes needed to update.”. * The system saves the changes and displays a success message indicating that the profile information has been updated. | |

## Module 2: E-commerce Related

|  |  |
| --- | --- |
| ID: TC04 | Test Case: Verify Products Listing |
| Purpose: To ensure that products are listed correctly on the e-commerce platform. | |
| Expected Results:   * When the user first enters the product page, it retrieves data from local storage and displays a list of products. * The system should display relevant products matching the search query. | |
| Actual Results:   * The products display corresponding auto-generated images related to the package title. * A flashing border will appear on the relevant products and hide after 5 seconds. | |

|  |  |
| --- | --- |
| ID: TC05 | Test Case: Verify adding products to the shopping cart |
| Purpose: To ensure that users can add products to their shopping cart. | |
| Expected Results:   * The products should be added to the shopping cart when you click on the “Add to Cartbutton. * The quantity of the product should change when the user modifies the quantity of the product. | |
| Actual Results:   * The system adds the product to the user's shopping cart. * The system records the newest quantity of the product and stores it in local storage. | |

|  |  |
| --- | --- |
| ID: TC06 | Test Case: Verify if maximum stocks have apply properly |
| Purpose: To ensure that the maximum stock limit is enforced correctly for a product. | |
| Expected Results:   * When users modify the product beyond the maximum stock number, the system should prevent further additions. | |
| Actual Results:   * The system displayed the error message “Maximum order is x”. | |

|  |  |
| --- | --- |
| ID: TC07 | Test Case: Verify checkout process |
| Purpose: To ensure that users can successfully complete the checkout process with a seamless experience. | |
| Expected Results:   * When a user clicks on the “Process to Payment ”button with no products selected, the system should prevent further action with messages to notify the user. * When a user clicks on the “Process to Payment ”button with a non-empty cart, redirect the user to the appropriate page to process payment. | |
| Actual Results:   * The system displayed the error message “No items selected” and closed the payment details modal to let users select products. * The user was redirected to the payment page for the checkout process. | |

|  |  |
| --- | --- |
| ID: TC08 | Test Case: Validate user inputs for payment detail |
| Purpose: To ensure that user inputs for payment details are properly validated. | |
| Expected Results:   * The system should display an error message when the user leaves any input fields empty. * The user successfully paid their transaction, and the data should be stored back on the payment record page. | |
| Actual Results:   * The system displayed the error message “Please fill in all fields” to prevent further action from being triggered. * The system shows a success message and stores data in local storage for later records. After 3 seconds, redirect the user back to the home page. | |

|  |  |
| --- | --- |
| ID: TC09 | Test Case: Verify payment record functionalities |
| Purpose: To ensure that payment record functionalities are working as expected. | |
| Expected Results:   * The payment record page should display the relevant information for each payment transaction when the user has a history of payment records. * When no record is found, the page should remain empty with a notification message. | |
| Actual Results:   * The system displays the relevant information for each payment transaction, such as payment amount, date, payment method, and status. * The system displays “No transaction found”and provides a button for quick access to the product page. | |

## Module 3: Comment Page and Reply System

|  |  |
| --- | --- |
| ID: TC10 | Test Case: Verify commenting system functionalities |
| Purpose: To ensure that the commenting system works as expected. | |
| Expected Results:   * When a visitor tries to post a comment, the system prompts the user to login first. * The logged-in user posts a new comment to a specific blog successfully. * The logged-in user replies to an existing comment from a specific blog successfully. | |
| Actual Results:   * The system prompts the user to log in before posting a comment and redirects them to the login page. * The system accepts the comment input, associates it with the specific blog, and displays it in the comment section. * The system provides an option to reply to an existing comment and allows the user to enter their reply. | |

# Part VIII: Description for gathering ideas-tool

For this project, we will leverage a combination of GitHub, Google Drive, WhatsApp, and Kanban to foster efficient collaboration and effective communication.

GitHub will serve as our central platform for version control, issue tracking, and code collaboration. Through branches and pull requests, team members can work on different features or bug fixes and merge changes into the main branch.

Google Drive will provide us with a shared folder to collaborate on project-related documents. Team members can work simultaneously on project plans, requirements, and design documents, making real-time edits and providing feedback.

WhatsApp will be our communication hub, enabling real-time discussions and idea sharing among team members. We can gather project ideas, communicate updates, and exchange files and code snippets through group chats.

In addition to these tools, we will incorporate a Kanban board to enhance project management. By utilizing a Kanban tool like Jira, we can visualize our workflow and track tasks effectively. Team members can create Kanban cards for tasks or user stories, assign themselves, set due dates, and move cards across columns to reflect progress. This enables us to manage work in a visual and transparent manner, ensuring a smooth workflow and prioritizing tasks accordingly.

# Part IX: Stakeholders Analysis

## Internal Stakeholders

|  |  |
| --- | --- |
| **Stakeholders** | **Goals to achieve** |
| Project Sponsor | * Provide financial support throughout the project lifecycle * Make decisions to ensure the project outcome |
| Project Manager | * Leader to responsible for guiding the team * Monitor the progress ensure meet customer’s expectations |
| Business Analyst | * Gather and analyze business requirements * Improve processes and deliver value to the organization |
| Developers and programmers | * Develop the software on different layers of software development * Analyze and debug any technical issues |

## External Stakeholders

|  |  |
| --- | --- |
| **Stakeholders** | **Goals to achieve** |
| Traveler | * Primary user to communicate and interact with the system * Ensure proper functionality and optimal performance |
| Website platform owner | * Manage and maintain the blogs and comment to ensure the quality of the website |
| Membership Members | * Encourage to actively engage and participate in the activities and initiatives of the system * Gain a collective feedback and representation to help improve the system features and experience continuously |

# Part X: Roles and responsibilities

| Name | Student Number | Role | Position | Contribution |
| --- | --- | --- | --- | --- |
| Tam Hin Pak | 5976807 | Project Manager | Project Manager | contributed |
| Chan Lok Ching | 8563196 | Team Member | Full-stack Developer | contributed |
| Chu Kai Chit | 8711136 | Team Member | Business Analyst | contributed |

# 

# 

# 

# 

# 

# 

# 

# Appendix

## Meeting minutes

First system meeting - minutes

Date: 27 December 2023

Time: 9:00 p.m. to 12:00a.m.

Location: Discord Online Meeting Room

Agenda Items

1. The team discussed the overall project flow, including the project scope and deliverables. Everyone generally collaborates to work on it.
2. The team identified the potential stakeholders involved in the system. Woody highlighted some external stakeholders that should be considered in the project scope and planning.

Second system meeting - minutes

Date: January 4, 2024

Time: 9:00 p.m. to 12:00a.m.

Location: Discord Online Meeting Room

Agenda Items

1. The team focused on gathering user requirements for the system, specifically related to participants and event hosts. Each team member suggested different functional and non-functional requirements based on their understanding and expertise.

Third system meeting - minutes

Date: January 15, 2024

Time: 9:00 p.m. to 12:00a.m.

Location: Discord Online Meeting Room

Agenda Items

1. The team collaboratively created a detailed project timeline, breaking down the tasks and activities required for each sprint using ProjectLibre. Paco finalized the project timeline and distributed it among the team members.
2. All team members work together to complete the project charter and requirements gathering.

Fourth system meeting - minutes

Date: February 1, 2024

Time: 9:00 p.m. to 12:00a.m.

Location: Discord Online Meeting Room

Agenda Items

1. We start to work on the prototyping of the system. The functionalities, user interface, and overall user experience were showcased to gather feedback and suggestions for improvement.
2. Based on the feedback received, the team developed an action plan for subsequent sprints of the prototype.

Fifth system meeting - minutes

Date: February 28, 2024

Time: 9:00 p.m. to 12:00a.m.

Location: Discord Online Meeting Room

Agenda Items

1. The team discussed the need for better version control of the project's prototype. It was agreed upon that utilizing GitHub, a widely adopted version control platform
2. The team discussed the importance of maintaining up-to-date documentation and a comprehensive Readme file in the GitHub repository.

Sixth system meeting - minutes

Date: March 15, 2024

Time: 9:00 p.m. to 11:00p.m.

Location: Google Meet

Agenda Items

1. We design various test cases to fit our quality assurance.
2. The prototype is nearly complete. The team started to perform multiple tests, including but not limited to stress tests, user acceptance tests and test cases.

Seventh system meeting - minutes

Date: March 17, 2024

Time: 1:00 p.m. to 6:00p.m.

Location: Discord Online Meeting Room

Agenda Items

1. Final review of the report and prepare for the prototyping presentation.

## Source Code

We have posted our project prototype on GitHub to leverage its powerful version control capabilities and facilitate collaboration among team members. At the same time, we have provided the source code folder in the Moodle submissi.

Link: https://andy-clc.github.io/Travel-Platform/