| Fontys University of Applied Sciences

![A bus on the road

Description automatically generated]()

**Himalayan Bus****: Project Plan**

Bus tickets reservation system

Himal Aryal [2954222]

2023

Table of Contents

[1. Project assignment 3](#_Toc146184902)

[1.1 Context 3](#_Toc146184903)

[1.2 Goal of the project 3](#_Toc146184904)

[1.3 Scope and preconditions 3](#_Toc146184905)

[1.3.1 Inside Scope: 3](#_Toc146184906)

[1.3.2 Outside Scope: 3](#_Toc146184907)

[1.4 Strategy 4](#_Toc146184908)

[1.5 Research questions and methodology 4](#_Toc146184909)

[1.5.1 Research Question 1: 4](#_Toc146184910)

[Methodology (Dot Framework): 4](#_Toc146184911)

[1.5.2 Research Question 2: 4](#_Toc146184912)

[Methodology (Dot Framework): 4](#_Toc146184913)

[1.5.3 Research Question 3: 5](#_Toc146184914)

[1.6 End products 6](#_Toc146184915)

[2. Activities and time plan 7](#_Toc146184916)

[2.1 Phases of the project 7](#_Toc146184917)

[2.1.1 Project Initiation Phase: 7](#_Toc146184918)

[2.1.2 System Design and Planning Phase: 7](#_Toc146184919)

[2.1.3 Development and Implementation Phase: 7](#_Toc146184920)

[2.1.4 Testing and Quality Assurance Phase: 8](#_Toc146184921)

[2.1.5 Deployment and User Training Phase: 8](#_Toc146184922)

[2.1.6 Operations and Monitoring Phase: 8](#_Toc146184923)

[2.1.7 Data Analysis and Decision-Making Phase: 8](#_Toc146184924)

[2.1.8 Project Handover Phase: 8](#_Toc146184925)

[2.1.9 Evaluation, Reflection, and Closure Phase: 8](#_Toc146184926)

[2.2 Time plan and milestones 8](#_Toc146184927)

[2.2.1 Sprint 1 8](#_Toc146184928)

[2.2.2 Sprint 2 9](#_Toc146184929)

[2.2.3 Sprint 3 9](#_Toc146184930)

[2.2.4 Sprint 4 9](#_Toc146184931)

[2.2.5 Sprint 5 9](#_Toc146184932)

[2.2.6 Sprint 6 9](#_Toc146184933)

[3. Testing strategy and configuration management 11](#_Toc146184934)

[3.1 Testing strategy 11](#_Toc146184935)

[3.2 Test environment and required resources. 12](#_Toc146184936)

[3.3 Configuration management 12](#_Toc146184937)

[4. Risk 14](#_Toc146184938)

[4.1 Risk and mitigation 14](#_Toc146184939)

**Version history**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Version** | **Date** | **Author(s)** | **Changes** | **State** |
| 1 |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

# Project assignment

## Context

"Himalayan Bus" is a bus service provider operating in a European region. They are committed to improving their services and customer experience by embracing technology.

## Goal of the project

The goal of the bus tickets reservation system (“Himalayan Bus”) project is to create an efficient online platform for booking and purchasing bus tickets. This project aims to provide customers with a user-friendly experience, optimize the bus service provider's operations, and generate revenue by leveraging Information and Communication Technology (ICT). The system adds value by enhancing customer satisfaction, improving efficiency, and fostering data-driven decision-making, all while positioning the company as a competitive and customer-centric service provider.

## Scope and preconditions

### Inside Scope:

1. User Profile Management: Within the project's scope, user profile management includes user registration, login, and basic user information maintenance. Users can update their personal details, view their booking history, and reset their passwords.
2. Bus Route and Schedule Management: The project includes features that allow administrators to manage bus details, such as Bus Name, Bus Route, Bus Type (AC/non-AC), seats, departure time, and arrival time. Admins can also manage the available routes.
3. Feedback and Reviews: Gathering and displaying user feedback and reviews for bus journeys is part of the project's scope. Users can provide feedback on their travel experiences, and this information will be accessible to other users for reference.

### Outside Scope:

1. Payment Gateway Integration: The integration of specific payment gateways for processing ticket payments is typically outside the initial project scope. This integration involves financial transactions and requires careful consideration of security and compliance.
2. Real-Time Bus Tracking: Implementing real-time tracking of buses and providing live updates on bus locations and delays is typically a complex and resource-intensive feature that is often considered for future enhancements rather than the initial project.
3. Advanced Analytics and Reporting: In-depth analytics and reporting functionalities, such as complex data analysis, revenue forecasting, or route optimization based on historical data, may fall outside the initial project scope. These advanced features often require more extensive development and data processing.

## Strategy

The Agile approach is a strong fit for the bus ticket reservation project for several reasons:

1. Changing Requirements: In a project like this, requirements can evolve as customer needs and market conditions change. Agile allows for flexibility in accommodating these changes without disrupting the entire project plan.
2. User-Centric Focus: The project's success depends on providing a user-friendly and efficient platform. Agile's emphasis on customer feedback ensures that user preferences are incorporated into the design and functionality of the system.
3. Incremental Development: Agile enables the delivery of working features in short cycles. This means that even in the early stages of the project, functional components can be delivered to users, providing value and allowing for early validation.
4. Collaboration: The bus ticket reservation system involves various stakeholders, including developers, administrators, and end-users. Agile promotes collaboration and regular communication among these groups to ensure everyone's needs are addressed.
5. Continuous Improvement: Agile's iterative approach allows the team to learn from each sprint and make improvements, enhancing the quality of the system over time.

In summary, the Agile approach is a well-suited strategy for the bus ticket reservation project because it aligns with the project's need for adaptability, user-centric design, incremental development, collaboration, and continuous improvement. It provides a framework that can accommodate changing requirements and deliver a more successful and customer-focused product.

## Research questions and methodology

### Research Question 1:

How can we enhance the user experience to make the ticket booking process more intuitive and user-friendly?

### Methodology (Dot Framework):

* **D**esign Thinking Workshops: Conduct workshops with users and stakeholders to gather feedback on the current booking process and ideate improvements.
* **O**nline Surveys: Administer online surveys to collect quantitative data on user preferences and pain points.
* **T**esting and Prototyping: Create prototypes of the booking interface and conduct usability testing with real users to identify areas of improvement.
* **A**nalytics and User Behavior Analysis: Use web analytics to track user behavior on the platform and identify drop-off points or bottlenecks in the booking process.

### Research Question 2:

What are the most popular bus routes and preferred bus types among users?

### Methodology (Dot Framework):

* Data Analysis: Analyze historical booking data to determine which routes are the most frequently booked and which bus types (AC/non-AC) are preferred.
* Online Surveys: Include questions in online surveys that specifically ask users about their preferred routes and bus types.
* Trend Analysis: Monitor and compare booking trends over time to identify shifts in user preferences.
* A/B Testing: Conduct A/B tests on the platform to evaluate the popularity of different bus types and routes.

### Research Question 3:

How satisfied are users with the provided bus services, and what are the main factors affecting their satisfaction?

Methodology (Dot Framework):

* Deep User Interviews: Conduct in-depth interviews with a sample of users to gather qualitative insights into their satisfaction levels and the factors that influence their experience.
* Online Surveys: Administer satisfaction surveys to collect quantitative data on user satisfaction levels.
* Text Analysis: Analyze feedback and reviews provided by users to extract insights on what aspects of the bus service are praised or criticized.
* Analytics: Use analytics to track customer support tickets and complaints, identifying recurring issues that impact satisfaction.

## End products

A screenshot of a computer screen

Description automatically generated

# Activities and time plan

## Phases of the project

In the case of the "Himalayan Bus" bus ticket reservation system project, here are the main phases:

### Project Initiation Phase:

* 1. **Problem Analysis and Requirements Gathering**:
     + Identify the challenges in the current bus ticket reservation process.
     + Define the specific requirements for user and admin authentication, bus details management, route management, seat confirmation, user registration, ticket booking, feedback submission, and reservation cancellation.

### System Design and Planning Phase:

* 1. **System Architecture and Database Design**:
     + Design the system architecture to accommodate user and admin authentication using session UUID.
     + Plan the database schema to store bus details, routes, user information, and reservations.
  2. **User Interface Design**:
     + Create user-friendly interfaces for both admins and users to interact with the system.

### Development and Implementation Phase:

* 1. **Authentication Module**:
     + Develop the authentication module for users and admins using session UUID for secure access.
  2. **Admin Features Implementation**:
     + Implement features for admins to add bus details, including Bus Name, Bus Route, Bus Type, seats, departure time, and arrival time.
     + Develop functionality for admins to define and manage bus routes.
     + Enable admins to confirm seat reservations for users.
  3. **User Features Implementation**:
     + Develop user registration functionality, allowing users to provide their details.
     + Implement the ticket booking system, enabling users to select source and destination.
     + Create a feedback submission module for users to provide their bus travel experiences.
     + Develop the reservation cancellation feature for users.

### Testing and Quality Assurance Phase:

* 1. Perform comprehensive testing to ensure the system is secure, functional, and user-friendly.
  2. Address any bugs or issues identified during testing.

### Deployment and User Training Phase:

* 1. Deploy the system to a production environment.
  2. Provide training to administrators and support staff for system management.
  3. Educate users on how to use the platform for ticket booking and other features.

### Operations and Monitoring Phase:

* 1. Continuously monitor the system for performance and security.
  2. Address user inquiries and issues promptly.
  3. Optimize the system as needed based on user feedback and operational data.

### Data Analysis and Decision-Making Phase:

* 1. Collect and analyze data on user behavior, booking patterns, and feedback.
  2. Use insights to make data-driven decisions for system improvements and business strategies.

### Project Handover Phase:

* 1. Prepare comprehensive documentation for the system, including user manuals and maintenance guides.
  2. Transfer system management responsibilities to the appropriate teams.
  3. Ensure a smooth handover of the project.

### Evaluation, Reflection, and Closure Phase:

* 1. Evaluate the project's success based on predefined criteria and KPIs.
  2. Reflect on the project's challenges, successes, and areas for improvement.
  3. Officially close the project, complete financial and administrative tasks, and archive project documentation.

## Time plan and milestones

The artifacts listed in each sprint represent the work items to be completed during that sprint. Let's review each sprint's artifacts:

### Sprint 1

1. Project plan.
2. Initial product backlog.
3. Backend: first setup of RESTful API.
4. Layering, dependency inversion and injection using Spring.
5. CI/CD environment initialization.

### Sprint 2

1. Architecture constraints and design decisions.
2. C4 Model diagrams with context/explanations for level 1 (system context), 2 (container) and 3 (component).
3. Backend: CORS configuration in-place for controllers.
4. Initial Frontend setup.

### Sprint 3

1. Design document version 2.
2. Initial Backend to Database setup.
3. SonarQube.

### Sprint 4

1. Design document version 3.
2. Authentication and authorization implementation.
3. Continuous Integration and SonarQube.

### Sprint 5

1. Final design document.
2. WebSocket’s feature.
3. Minimum viable product (MVP) features implementation.
4. Continuous Integration and SonarQube.

### Sprint 6

1. Final UX feedback report
2. Final individual track product with minimum viable product (MVP) features implemented.
3. Continuous Integration and SonarQube.
4. Continuous Delivery.
5. Hand in all your work.

|  |  |  |  |
| --- | --- | --- | --- |
| **Phasing** | **Effort** | **Start date** | **Finish date** |
| Sprint 1 | High | 9/15/2023 | 9/22/2023 |
| Sprint 2 |  |  |  |
| Sprint 3 |  |  |  |
| Sprint 4 |  |  |  |
| Sprint 5 |  |  |  |
| Sprint 6 |  |  |  |

# Testing strategy and configuration management

## Testing strategy

A diagram of testing

Description automatically generatedIn the testing strategy for the bus ticket reservation system project, I will employ a multi-level approach to ensure the reliability and quality of the software. The following testing levels will take place:

**Unit Testing:**

* **Scope:** Unit testing will be performed on individual components and functions of the system, such as specific modules, classes, or methods.
* **Automation:** Unit tests will be automated using testing frameworks like JUnit or NUnit.
* **Goals:** Achieve high code coverage (e.g., 80% or higher) for unit tests to ensure that critical components are thoroughly tested.

**Integration Testing:**

* **Scope:** Integration testing validates the interactions between different parts of the system, including the frontend and backend.
* **Automation:** Some integration tests can be automated, especially those involving APIs or data integration.
* **Goals:** Identify and address any issues related to data exchange, communication, and overall system integration.

**System Testing:**

* **Scope:** System testing evaluates the entire bus ticket reservation system as a whole. It focuses on end-to-end scenarios and user journeys.
* **Automation:** While many system tests will be manual to simulate real user interactions, some regression tests can be automated.
* **Goals:** Ensure that the system meets all functional and non-functional requirements, including usability, performance, and security.

**Acceptance Testing:**

* **Scope:** AT involves real users (or simulated users) testing the system to determine if it meets their needs and expectations.
* **Automation:** Typically, UAT is performed manually by users to mimic their experience realistically.
* **Goals:** Obtain user feedback and ensure that the system aligns with user requirements.

**Justification:**

* **Comprehensive Coverage:** This multi-level testing strategy ensures comprehensive coverage, from unit-level validation to end-to-end system testing.
* **Early Issue Detection:** Unit and component testing help detect and address issues at an early stage of development, reducing the cost of fixing defects later.
* **User-Centric Approach:** User acceptance testing ensures that the system meets user expectations and is aligned with real-world usage.
* **Non-functional Validation:** Non-functional testing guarantees that the system not only functions correctly but also perform well, is secure, and can scale as needed.
* **Automation Benefits:** Automation helps in regression testing and ensures that code changes do not introduce new defects.
* **Flexibility:** Manual testing allows for exploratory testing and the identification of unexpected issues.

## Test environment and required resources.

**Test Environment:**

A diagram of a testing process

Description automatically generatedFor the bus ticket reservation system project, I envision a test environment that follows a typical DTAP (Development, Testing, Acceptance, Production) pipeline. This approach ensures that each phase of the project is thoroughly tested before moving to the next stage.

**CI/CD Pipeline:**

To streamline development and testing, a Continuous Integration/Continuous Deployment (CI/CD) pipeline will be established. The CI/CD pipeline automates the build, testing, and deployment processes. It helps in ensuring that code changes are integrated and tested frequently, leading to faster development cycles and more reliable releases.

## Configuration management

In the bus ticket reservation system project, a robust configuration management approach will be implemented using Git as the version control system. Here are the key aspects of the version management strategy:

**Tooling:** Git will be used as the primary version control system, and a central Git repository will be set up to manage the project's source code and related files. Git provides powerful branching, merging, and tracking capabilities.

**Branching Strategy (Single Branch - master):**  
In a simplified individual project, using only one branch (typically the **master** branch) is a straightforward approach.

**Master Branch:** The **master** branch serves as the primary branch for my project. All development work, including new features, bug fixes, and improvements, is done directly in this branch.

**Promotion, Release, and Baseline Strategy:**

* **Promotion:** In this simplified approach, code moves progressively forward within the master branch, from development to stable and production-ready states.
* **Release Strategy:** For significant milestones or versions, I can create version tags within the main branch to mark specific releases. These tags act as snapshots of my project's state at those points in time.
* **Baseline Strategy:** While not formally designated as baselines, my version tags effectively serve as baselines for tracking the project's progression.

# Risk

## Risk and mitigation

|  |  |  |  |
| --- | --- | --- | --- |
| **Risk** | **Impact** | **Prevention activities** | **Mitigation activities** |
| 1. **Technical Challenges** | High | * Conduct thorough research and prototyping before starting development. * Seek advice from online communities and experts when facing technical difficulties. | * Allocate extra time in the project schedule to account for unforeseen technical challenges. * Maintain a list of alternative solutions or approaches in case of technical roadblocks. |
| 1. **Scope Creep** | Moderate | * Clearly define and document the project's scope and objectives. * Establish a change control process for scope changes. | * Review and assess all proposed scope changes before implementation. * Communicate with stakeholders to manage expectations when scope changes are approved. |
| 1. **Time Constraints** | Moderate | * Develop a realistic project schedule with buffer time for unexpected delays. * Prioritize tasks based on their criticality and dependencies. | * Regularly monitor progress against the project schedule and adjust timelines as needed. * Consider adjusting non-critical project tasks or features if schedule constraints become apparent. |