

Project name: BUS RESERVATION SYSTEM

Name: NICOLETAH NJERI KAGURU

Reg: 20/03930

Unit code: BSD 3106

Unit name: Final year 1 project proposal

A PROJECT PROPOSAL SUBMITTED IN

PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE AWARD OF

INFORMATION TECHNOLOGY IN THE SCHOOL OF TECHNOLOGY AT KCA

UNIVERSITY

DECLARATION

I declare that this dissertation is my original work and has not been previously published or submitted elsewhere for award of a degree. I also declare that this contains no material written or published by other people except where due reference is made and author duly acknowledged.

NAME: NICOLETAH NJERI

REG NUMBER: 20/03930

SIGNATURE:

DATE:

ABSTRACT

The main objective of the study was to provide better work efficiency, accuracy, reliability in the transport network. This helps them to access buses easily to their preferred destinations. Each user will receive a reference number after successful reservation of a seat on the given fleet of buses. The goal of this study is to cater for the needs of a traveler by providing luxurious buses, with efficient bus booking system. There are plenty of luxurious buses available now, providing best & efficient services to the customers. The result of the system is that will provide easier access to transport modes easily and be assured of a seat compared to where one needs to board a matatu. The system keeps track of total tickets sold for each schedule. To be more precise the system helps to keep track of the bus schedule, their management, and also displays all the available bookings. This is done by the administrator who is the controller of the system. If the buses are available, the user can reserve seats by providing details such as name and ticket quantity. Each user will receive an order reference number after each successful reservation.

ACKNOWLEDGEMENT

First the researcher would like to thank God, our provider and strength, for the ideas, knowledge and capability to progress this far with this project and this studies as a whole. Much appreciation goes to this supervisor, Ezekiel Kuria for his guidance, support, and invaluable advice for the duration of this unit. The researcher would also like to appreciate his peers and classmates for their criticism and support without which this proposal would not be a reality. Last but not least, he would like to thank his family and friends for the moral support and encouragement they have offered him.

DECLARATION	2
ABSTRACT	2
ACKNOWLEDGEMENT	3
1. Chapter I: Introduction:	10
1.1 Background:	10
1.2 Problem Statement:	10
1.3 Proposed system.	10
1.4. Project objectives:	11
2. Chapter II: Literature review:	12
2.0 Literature review	12
2.1 Regional application of online bus reservation system	12
2.3 Regional Analysis of Online Bus Reservation Systems in Nigeria	12
An assessment of the strengths and weakness of the existing Projects Strengths;	14
2.3 Establishing the need for a current project.	15
Chapter III: Methodology	16
3.1 INTRODUCTION:	16
This chapter outlines the research methodology employed, detailing the techniques utilized and the system requirements.	16
3.2 Project Locality and Beneficiaries:	16
3.2 Target Group:	16
3.3 Sampling Method:	16
3.4 System analysis and design	17
3.4.1 Software Development	17
3.4.2 Extreme Programming	18
BUDGET	20
Proposed Budget	20
PROJECT SCHEDULE	21
2.0 SYSTEMS REQUIREMENTS	22
2.1 INTRODUCTION	22

INTENDED	22
Project	22
Overall description	22
User classes and characteristics	22
2.2 Operating	23
Hardware	23
1. Hardware- the system is designed to be compatible with various platforms, including desktop computers, laptops, tablets and smartphones. It should run on both Windows ensuring flexibility and accessibility	23
SOFTWARE	23
2.3 DESIGN&IMPLEMENTATION	24
Assumptions	24
Bus Reservation System	25
Reservation Management	26
2.4 EXTERNAL INTERFACEREQUIREMENTS.	27
I. User	27
lii. Software	28
iv. Communication	29
2.5 Non-functional	29
A. Performance	29
B) Safety	30
C) Security Requirements	31
The system should comply with data implement measures to safeguard travelers' data, handle access requests, clear privacy External Policies and Regulation: The system must comply with industry regulations and guidelines related to data security	31
Other Requirements	31
3.0 SOFTWARE DESIGN SPECIFICATION	33
3.1 System Overview	33
3.2 SYSTEM ARCHITECTURE	34
• Presentation Layer:	34

li. Application Layer:	34
lii. Data Layer:	34
Iv. Hardware Components:	34
V. Network Infrastructure:	34
3.3 Software Design	35
Logical Design	35
Use Case Diagram	35
3.3.1 RESERVATION MODULE	36
File and Database Design	38
3.5 Database Description	38
3.5.1 ADMIN TABLE	38
3.5.2 AVAILABLE CLASS TABLE	39
3.5.3 BOOKING DETAILS	39
3.5.4 TICKETS GENERATED TABLE	40
3.5.5 TRANSACTIONS TABLE	40
3.6 ENTITY RELATIONSHIP DIAGRAM	41
3.7 System Integrity controls	42
4.0 TEST PLAN INTRODUCTION	43
4.1 Goals and objectives	43
4.2 Statement of Scope	43
4.3 Major constraints	44
4.4 Software to be tested	44
4.4.1 Testing strategy	44
A) Unit testing	44
B) Integration testing	45
C) Validation testing	45
D) High- Order testing	45
4.5 Testing resources and staffing	46
4.6 Test work products	46

4.7 Test Record Keeping	46
4.8 Test Metrics	47
Test Procedure	48
Software to be tested	48
4.9.1 Test Cases for Component	48
Test case 1: Booking info & travelling	48
Test case 2: Contact Info	49
Test case 3: Payment & verification	50
Test case 4: Verify order details	50
4.9.2 Purpose of Tests for Component	50
Expected results for Component	51
Integration Testing	51
Stubs and drivers required:	51
Test Cases and their Purpose: -	51
Expected Results:	51
Validation testing	52
High-Order testing (System testing)	52
Staffing:	53
Test Work Products	53
Test record keeping and record	54
5.0 SYSTEM IMPLEMENTATION	55
5.1 INTRODUCTION	55
PURPOSE OF IMPLEMENTATION	55
PROPOSED SYSTEM	55
5.2 INSTALLATION & CONVERSION PLAN	55
5.3 IMPLEMENTATION REQUIREMENTS	56
TOOLS USED	56
5.4 USER INTERFACE DESIGN IMPLEMENTATION	56
1. Generating order reference	56

2. Travelling and booking info	57
3. Contact info	57
4. Payment and verification	58
5. Confirm details	58
6. Admin Access Panel	59
5.5 OUTPUT OF THE SYSTEM	59
HARDWARE AND SOFTWARE NEEDED	60
5.6 TRAINING OF USERS	61
SYSTEM LIMITATIONS	61
RECOMMENDATIONS	61
5.7 REFERENCES	62

1. Chapter I: Introduction:

1.1 Background:

Kimathi Bus Station is located adjacent to the Kimathi post office and the former District Office Headquarters, serves as a central hub for various transportation services within the area. It hosts multiple terminals facilitating travel to neighboring towns such as Nyeri, Nakuru, and Meru, as well as to smaller satellite towns. To streamline operations and ensure a steady income flow among investors, many vehicle owners have leased their buses and matatus to different cooperative societies (Saccos). Its aimed to provide efficient and flexible transportation to clients within Kimathi.

1.2 Problem Statement:

The existing challenge arises when commuters need to catch a matatu or bus at the station, requiring them to arrive well in advance to secure a seat for their desired destination. This early arrival is essential, particularly for longer journeys, to ensure a seat and to stay informed about fluctuating fare prices. However, the unreliability of bus arrivals creates uncertainty among passengers, leading them to question whether to wait or seek alternative transportation options. Furthermore, the manual ticketing system is prone to errors, resulting in some passengers traveling without valid tickets. Additionally, long queues and the necessity to pay in cash further exacerbate the inconvenience for customers seeking to acquire bus tickets.

1.3 Proposed system.

The proposed system is a user-friendly application designed to facilitate online bus ticket reservations for commuters. Through this system, commuters can easily reserve seats for their preferred destinations by providing necessary details such as their name, travel destination, preferred bus class, payment information, and the number of seats required. Upon successful reservation, a unique reference number is generated for each transaction, allowing users to reprint tickets or generate new ones if needed.

The system also grants administrators full control, enabling them to manage reservations by accessing comprehensive details such as commuter names, reference numbers, amounts paid, and destinations. Administrators can also reject reservations if necessary. Moreover, the system efficiently tracks the total number of tickets sold for each schedule and provides warnings when seats are fully occupied. Additionally, it maintains records of available locations and manages transport-related data.

1.4. Project objectives:

- Streamlining the ticket booking process, formerly burdened by extensive paperwork, and ensuring thorough documentation of essential information.
- Aims to simplify, enhance reliability, and improve user experience of the Kimathi bus reservation system, making it intuitive and error-correcting.
- Offers features such as seat reservation, cancellation, and various inquiries, catering to the need for swift and immediate booking.
- Intends to create tailored software solutions for online ticket booking.
- Admin privileges include updating and canceling payments, as well as managing route reservations.

2. Chapter II: Literature review:

2.0 Literature review

This is an application that is accessed by users over a network such as the internet or an intranet.

2.1 Regional application of online bus reservation system

A Case Study in the United States

In America (2015), the implementation of an online bus reservation system in the United States was explored. The system aimed to streamline the booking process for users, allowing them to reserve seats with ease and convenience. This emphasized the significance of such systems in modernizing the transportation sector, reducing waiting times, and enhancing customer satisfaction. They highlighted the versatility of the technology utilized, suggesting its potential applicability to other industries beyond transportation. Additionally, the study underscored the importance of user-friendly interfaces and accessibility features to ensure inclusivity. Overall, the research supported the adoption of online bus reservation systems as a means to improve efficiency and convenience for both passengers and service providers.

2.3 Regional Analysis of Online Bus Reservation Systems in Nigeria

In Nigeria, the bus travel industry is experiencing significant growth, yet the manual reservation process proves cumbersome and time-consuming for commuters, often leading to long queues. Recognizing this challenge, the adoption of online bus ticket reservation systems has emerged as a solution, allowing travelers to conveniently check bus availability, purchase tickets, and make online payments.

According to Melissa (2007), the implementation of an online bus ticketing system offers a comprehensive solution for managing ticketing operations. This includes features such as

- capturing customer information (name, address, phone number, and email)
- providing a price list
- ranking bus operators
- displaying seating charts, managing payment information
- generating reports

The reservation system typically comprises three modules, as outlined by Melissa (2007):

1. Availability Inquiry Module: This module enables customers to check seat availability for a specific bus on a particular date.
2. Reservation Module: Here, customers can reserve tickets for their desired bus and date.
3. Cancellation Module: Finally, customers have the option to cancel their reserved tickets if necessary..

The integration of these modules into an online bus reservation system not only enhances convenience for commuters but also streamlines operations for bus operators, ultimately contributing to the efficiency and growth of the bus travel industry in Nigeria.

The gap in the application : adequate internet access is required to fill in the forms for reservations of seats .One needs to be ready for a surge of new customers who fill for the reservation for their destined destinations.

An assessment of the strengths and weakness of the existing Projects Strengths;

- The reservation system used saves a lot of time used compared to when one wants to board a bus using the old strategies in place.
- II. You can choose a seat of your choice from the sitting arrangement and pay only what you should pay and nothing more.
- III. There is convenience as a user can book online ticket and get their ticket at irrespective of time.

Weakness

1. No protection from misleading pirates who pose as the owners of the organization to give misleading information
2. The user might be offline hence end up not receiving the information
3. System will not allow customers to refund

2.3 Establishing the need for a current project.

Based on the gathered information from diverse sources, it becomes evident that there is a compelling necessity for the present project. This stems from the requirement for improved accessibility and swifter provision of bus services, particularly in contrast to the outdated system characterized by lengthy queues.

Chapter III: Methodology

3.1 INTRODUCTION:

This chapter outlines the research methodology employed, detailing the techniques utilized and the system requirements.

3.2 Project Locality and Beneficiaries:

The reservation system operates online as a web application, primarily catering to travelers destined for their respective destinations.

3.2 Target Group:

The project's target demographic consists of commuters seeking streamlined access to transportation services, eliminating the need for lengthy queues. Additionally, an administrator has website access, enabling them to manage bookings, reject reservations, and view all bookings made.

3.3 Sampling Method:

Various data collection techniques were employed:

- **Systematic Sampling Technique:** Interactions with commuters at the Kimathi bus terminus revealed a demand for a more convenient booking process from the comfort of their homes or offices. This method ensured thorough system competency assessment with easily accessible details.
- **Observation:** Regular commuting to the Kimathi bus station facilitated information gathering regarding the necessity of an online bus ticketing system, especially for long-distance journeys.
- **Interviews:** Conversations with commuters emphasized the preference for transitioning from the traditional practice of queuing to the convenience of reserving tickets online. This corroborated the need for the proposed system.
- **Questionnaires** The researcher gave out the questionnaires with specific questions to the people being interviewed. The questions in the questionnaires were:
 1. How frequently do you utilize bus transportation services?
 2. On average, how much time do you spend waiting in queues to book a bus ticket?
 3. Are you aware of online bus reservation systems?
 4. Have you ever used an online platform to book bus tickets?
 5. What are the main challenges you face when booking bus tickets through traditional methods?
 6. Would you prefer an online bus reservation system over traditional booking methods? Why or why not?
 7. What features would you expect from an ideal online bus reservation system?
 8. How likely are you to use an online bus reservation system if it were available?
 9. How important is it for you to have the option to cancel or modify your bus reservation online?
 10. Do you have any additional comments or suggestions for improving bus reservation services?

3.4 System analysis and design

3.4.1 Software Development

This is a framework that is used to structure, plan as well as control the process of developing an information system. There are few development models which we usually follow.

- Agile software development

- Extreme programming

- JAD (Joint Application Development) Waterfall

- WSDM (web semantic design method)

Here, I am going to choose **XP (Extreme programming)** methodology to develop this system.

3.4.2 Extreme Programming

Extreme Programming (XP) stands out among agile methodologies for its flexibility in accommodating changes to requirements during iterations, in contrast to methodologies like Scrum and Rad.

Why xp?

- **Flexibility in Requirements Changes:** XP allows for changes in requirements even during development iterations, providing adaptability in dynamic project environments. This flexibility is particularly beneficial when working on projects with evolving or uncertain requirements, a feature not always present in other agile methodologies like Scrum.
- **Emphasis on Continuous Testing:** XP promotes a rigorous approach to testing, with tests written before code implementation. This ensures that the code is thoroughly tested, reducing the likelihood of bugs and facilitating early bug detection. Other agile methodologies may not prioritize testing to the same extent, leading to potential quality issues.
- **Customer-Centric Approach:** XP places a strong emphasis on customer involvement throughout the development process. Regular feedback and collaboration with the customer help ensure that the final product aligns closely with their expectations and requirements. While other agile methodologies also involve customer collaboration, XP's focus on continuous customer feedback is particularly pronounced.

XP follows five phases:

1. **Planning:** The team engages with clients to comprehend the desired product, gathering user stories to establish the project's goal.
2. **Designing:** Developers use the user stories to architect the code, ensuring alignment with customer expectations. This involves selecting programming languages, environments, libraries, and frameworks for software development.
3. **Coding:** Code is written iteratively in small increments and rigorously tested to maintain high quality and promptly address any bugs.
4. **Testing:** Automated tests are written to verify that the code functions correctly, identifying potential bugs before software release.
5. **Listening:** Customer feedback is actively sought to improve the software further. This ensures that the software meets customer expectations and maintains high quality.

A) Coding

MySQL

This is a multi-thread, multi-user, SQL relational database server. Programming language that can access a MySQL database include C, C++, Java, PHP and Perl. MySQL runs on many different operating systems including transactions, SSL support, nested SELECT, ACID compliance and Query Caching.

PHP

This acronym stands for Hypertext Preprocessor. It is mainly used as a general purpose scripting language used to develop dynamic web content and can be embedded in HTML. PHP can be used as an alternative to Macromedia ColdFusion, ASP.NET/C#/VB.NET and the JSP/Java system. PHP is a fully defined language and can be used to develop Graphic User Interface Application.

APACHE HTTP SERVER

This is a free and open source cross-platform web server package, consisting mainly of the Apache HTTP server, MySQL database and interpreters for scripts written in the PHP and Perl programming language. XAMPP'S name is an acronym for X meaning cross-platform, Apache HTTP server, MySQL, PHP and Perl.

Web Application Technologies

The web application was chosen in this project because they are open source which implies that they are cheap to get since one just need to download them from the net. PHP is a rapid application development environment and is known for its ease of use and it enables most developers get involved with dynamic web development. MySQL has very fast database management system and is also easier to use than many other database systems.

Semantic UI

Used as the conceptual building block of the website as interactive interfaces have been used to design the structure of websites framework. Uses JQuery and CSS to build greater experience user interfaces.

BUDGET
Proposed Budget

BUDGET ITEM	QUANTITY	AVAILABILITY	ESTIMATED COST IN (KSHs)	ACTUAL COST IN (KSH)
Laptop	1	Readily available	50,000	50000
Windows 10		Readily available	Free	Free
Antivirus	1	Readily available	2000	2,000
PHP MySQL Net-beans IDE		Readily available	FREE	FREE
SUBTOTAL			52,000	52,000
SUBTOTAL			56,000	56,0000

PROJECT SCHEDULE

This project schedule outlines the various tasks, expected start and end dates for each phase of the Kimathi Bus Reservation System

Task number	Task description	Expected start date	Actual start date	Expected end date	Actual end date
1	Project initiation	1/05/24	5/05/24	9/05/24	9/05/24
2	Requirements gathering and analysis	15/06/24	15/06/24	23/06/24	23/6/24
3	Design and prototyping	5/7/24	6/7/24	10/9/24	15/9/24
4	Development	18/10/24	20/10/24	23/10/24	1/11/24
5	Documentation and training	5/11/24	9/11/24	23/11/24	25/11/24
6	Deployment and launch	2/12/24	2/12/24	7/12/24	7/12/24

2.0 SYSTEMS REQUIREMENTS

2.1 INTRODUCTION

The Kimathi Bus Reservation System is the product identified in this document. This the system requirements for the entire Bus Reservation System, encompassing both and backend components. The scope of this SRS include the complete range of required for self-service interactions with customers who select their destinations from Kimathi

INTENDED

The Bus Reservation System SRS document is intended for various stakeholders development, management, marketing, testing and documentation of the system. audience for this document include developers, users.

Project

The Bus Reservation System is a software solution designed to experience by providing a seamless and self-service approach to book seats destination. The purpose of this system is to enhance easier booking of passenger improving overall efficiency in the transport This is done through enabling destined travelers by reserving tickets by personal info, number of seats, payment methods to be able to get a bus. This reservation of tickets eliminates the need for lengthy queues and manual paperwork, a more efficient and convenient experience for The system aims to create a positive impression, increase customer loyalty positive response to customer's selection of desired

OverallDescription

Userclasses and characteristics

The bus reservation system is designed to cater to two user classes based on characteristics. User classes for this system can be differentiated based on security privileges, technical expertise. The following user classes

1. **Travelers**- these are the primary users of the system. No logins are required services of booking a bus. The traveler use the system to reserve tickets, also can be allowed to re-print
2. **Administrator**- these users have elevated privileges and access to the system's functions. They are responsible for maintaining the system, check all lists of reject reservations, and

confirm payments. Admin requires a higher level of manage the high number of payments and reservations made in order to .

3. **IT Support**- this user class consist of technical staff who provide assistance and any system-related issues that arise. They require a deep understanding of technical architecture, interfaces and integrations. IT support personnel play a ensuring the system operates smoothly and addressing any technical challenges- Travelers are the most favored user class as their satisfaction and user impacts the success and reputation of the system. It is also important to provide the training and support to administrators, IT support to ensure effective system technical

2.2 Operating

The Kimathi Bus Reservation system is designed to operate in a specific environment hardware platforms, operating systems and other software components.

Hardware

1. **Hardware**- the system is designed to be compatible with various platforms, including desktop computers, laptops, tablets and smartphones. It should run on both Windows ensuring flexibility and accessibility

SOFTWARE

The software should be compatible with popular operating as Windows 10, Android. The system should be tested and optimized for smooth these operating systems to ensure an enticing user experience across

Software

- **MySQL**- Programming language that accesses created databases for all This is a multi-thread, multi-user, SQL relational database
- **PHP**- Has been used to develop dynamic web content and can be embedded with PHP is a fully defined language and can be used to develop graphical
- **HTML & CSS**- Has been used in corporately to define the content and website Semantic
- **UI**- The interface elements mostly used; dropdowns, buttons has been by Semantic UI to create great user
- **APACHE HTTP SERVER**- open source cross-platform web server package, of the APACHE HTTP SERVER, MYSQL database and interpreters for scripts PHP

2.3 DESIGN&IMPLEMENTATION

The development of the Kimathi Bus Reservation System may be subject to certain limit the options available to the developers. These limitations could be :

1. **Hardware limitations-** The system development may face hardware limitations in timing requirement & memory usage. Software must be designed to operate within the available hardware resources to ensure optimal
2. **Security considerations-** Data security is of importance .the developer must security considerations such as access control, secure communication protection against
3. **Maintenance responsibility-** Proper maintenance of the system by ensuring and likely challages to occur in more use of the sytem.
- 4 **Regulatory policies-** The developer must adhere to corporate policies and requirements set by the transport industry. These policies may dictate data measures, privacy regulations, or specific protocols that need to be development

Assumptions

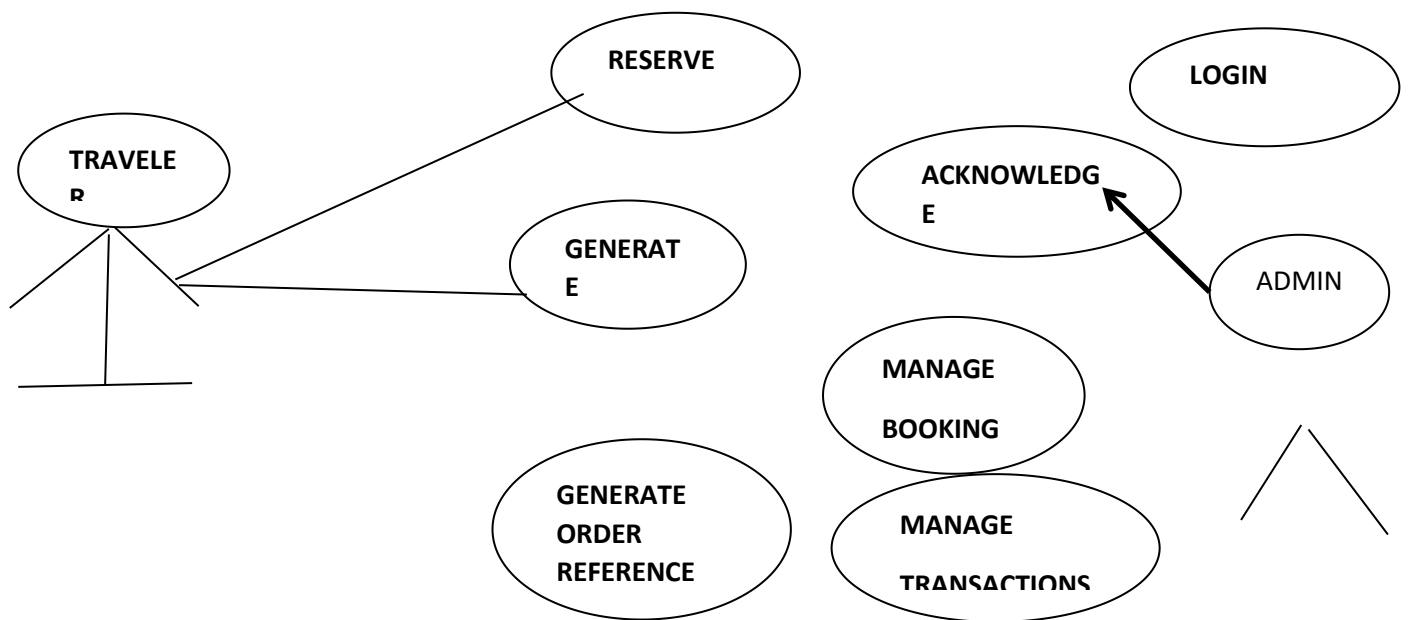
The following assumed factors could affect the requirements stated in the SRS Reservation Operating environment-

- ✓ The SRS assumes a certain development and environment, including specific versions of development tools, languages, operating systems and databases. Any updates to these could require modifications to the system's requirements to ensure and optimal
- ✓ Constraints and Limitations- SRS may make certain assumptions constraints and limitations, such as hardware resources, network regulatory compliance requirements.

If these assumptions are impact the system's requirements and necessitate adjustments to meet constraint of the System .

The Kimathi Bus Reservation System incorporates various system features efficient and convenient management of transport services. These organized based on their functionality and the user classes .

Following diagram is a usecase that represents the Kimathi Bus System



Here are some examples of system features.

Bus Reservation System

This is a crucial feature that allows travelers to book for seats for buses to destinations. It is of high priority as it directly contributes to enhancing customer and generate revenue .

Priority component.

Benefit :(Enables travelers to conveniently reserve seats, increase customer and company

Penalty :(minimal penalty associated with implementing and reservation

Cost :(moderate implementation and maintenance costs, including hosting updates

Risk :(potential risk of technical issues or booking errors if not properly handled.

Stimulus/Response User action: traveler clicks on order System Response: the system generates an order reference no identical file.

User action: traveler fills booking details requiring; destination, travelling date

- ◆ System Response: loads the filled details and proceeds to contact User action: traveler fills details containing; full name, contact/mobile/ e-mail gender then click to submit
- ◆ System Response: loads the filled details and proceeds to User action: traveler fills the payment method used, transaction ID, and amount to pay, clicks on
- ◆ System Response: processes the entered details and proceeds to confirm and details User action: traveler confirms the submitted information to confirm and get a ticket.
- ◆ System Response: processes and generates Kimathi Bus Reservation System streamlines the booking process for travelers, conveniently book seats of given bus destinations.

By following the provided sequence, the system ensures a smooth user experience and effectively handles process benefiting both the traveler.

Reservation Management

Responsible for efficiently managing the company's accounts reservations/bookings made, rejecting nonviable reservations. It is of high directly impacts the company's operations, revenue generation and satisfaction.

Priority Component

Benefit: (Optimizes seat allocation, reduce errors

Penalty: (Require training for understanding the system

Cost: (moderate implementation and maintenance

Risk: (maintenance issues if not properly handled.

Stimulus/Response

User action: admin logs in to the system with correct username & System Response: processes filled in details and loads

User action: admin manage all bookings or reject System Response: System deletes rejected The Reservation management and accounting feature ensures efficient company resources.

The system assist admins in managing Kimathi optimizing operation and maintaining a high standard of customer

2.4 EXTERNAL INTERFACEREQUIREMENTS.

I. User

In the Kimathi Bus Reservation System, the software interfaces with the various components, each with its own logical characteristics. These designed to provide a user-friendly and intuitive

Userinterface

- a) Order reference interface- this allows user to the generate order is used to list no of tickets
- b) Booking interface- allows users to book bus seat/(s), destination, No of and dates for their preferred
- c) Personal details interface- enables users to fill their personal details phone contacts
- d) Billing and payment interface- allows travelers to select the payment with the transactional code to be submitted together with the confirmed paid e. Confirm details interface- the selected details from the booking to interface are confirmed and ensures that all that is well.

Logical characteristics

Screen layout: the interfaces should follow a consistent and user-friendly layout navigation menus, clearly labeled buttons and organized information

GUI standards: the interfaces should adhere to established GUI standards consistency across system.

Screen Constraints: the interfaces should be designed to fit within the display size limitations of the target devices, ensuring optimal

ii Hardware

The System possess logical and physical characteristics associated with the the software product and the hardware. These characteristics define the types, data and control interactions used in the system.

logical characteristics

The software interacts with the hardware components, ensuring compatibility functioning. The physical characteristics focus on the actual device and technology

- a) Supported devices: the software interfaces should be designed to hardware devices commonly used by users, such as desktop computers
- b) Data and control interactions: the software interacts with the hardware perform tasks such as capturing user input, display information on the controlling peripheral devices. The logical characteristics define how data signals are exchanged between the software and
- c) Communication protocols: the interfaces may utilize standard communication to enable data transfer and interaction between the software and hardware. of communication protocols include Ethernet or

Physical characteristics of the System

- a) Input devices: the interface should support different input devices, such as mouse, and touchscreens allowing users to input
- b) Output devices: the software interfaces with output devices, including printers to present
- c) Data storage devices: the software may interact with physical storage hard drives to store and retrieve data related to the bus reservation

iii. Software

PHP is a programming language that has been used to develop bus It automates bus reservations and enables travelers to book for a seat to their destination. The components involved

1. **Database:** the Kimathi Bus Reservation System relies on MySQL database to store travelers' information, reservation details, billing and other

2. **Operating System:** PHP codes can run on various operating systems, including Desktop (e.g, Windows 10), or other compatible

3. **Integrated commercial Components:** Kimathi Bus Reservation System may external commercial components to enhance its functionality. For example, it could with payment gateways like M-PESA, banks to operate

a. Data items coming into the-traveler information: Name, contact details, identification-payment information: transaction

b. Data items going out of the-Reservation Confirmation: displayed on the screen when order is-Payment Receipts: displayed on the screen when ticket for a destination

iv. Communication

The Bus Reservation System may require various communication functions with travelers as they book for their destinations. Here the requirements :

i. Web browser

The system will include a web-based interface for travelers to make admins to manage bookings. It requires communication with web browsers using HTTPS for secure

ii. Electronic

The system will utilize forms to collect travelers' information during the Forms can be presented through the web interface. The form data can be HTTP POST requests or suitable protocols, depending on the system's

iii. Communication

The system will adopt communication standards such as HTTP for web-based FTP for file transfer with external systems. The choice of standards depends on communication requirements and integration with external

2.5 Non-functional

A. Performance

Performance requirements for Kimathi Bus Reservation System are ensure a smooth and efficient user experience. Here are some requirements that developers should

System Requirement:

The system should respond to user interactions within 1 Rationale: travelers using Kimathi Bus Reservation web interface responses when making bookings. A fast response time satisfaction

BookingProcessing Requirement.

The system should process booking request within 3 Rationale: travelers should be able make bookings without delays. This requirement ensures that the system can handle booking promptly

ConcurrentUser Requirement

the system should support a minimum of 100 users without significant degradation in Rationale: the system may experience a high volume of user Ensuring that the system can handle a sufficient number of users without performance

DatabaseResponse Requirement:

Database queries should return results within 1 Rationale: fast database response times are crucial to ensure retrieval and provide

real-time System Requirement:

the system should have an uptime of at least Rationale: the system should be available to travelers at all times, disruptions and ensuring reliable access to bus reservation services. can result in traveler dissatisfaction and potential

Data Requirement:

Updates to booked seats should be reflected across 5 Rationale: to provide accurate information to travelers and prevent bookings, the system should synchronize data in near real-time. Timely help maintain consistency across various interfaces and prevent-Payment Processing: Payment transactions should be authorized and processed seconds of the traveler's payment

B) Safety

The Kimathi Bus Reservation System should meet various requirements related to protection and compliance. Key points

1. Traveler data protection: The system should ensure security and confidentiality of data in compliance with data protection regulations and also implement encryption and access
2. Privacy Consent: The system should obtain consent from travelers regarding the and usage of their personal data, provide clear privacy policies, and allow data
3. Compliance with Regulations: the system should comply with local regulations related to safety, privacy and traveler protection and best
4. Safety Certifications: Depending on the jurisdiction and industry, the system satisfy safety certifications such as UL Certification or
5. Accessibility: The system should comply with accessibility standards, enabling travelers with disabilities through features like keyboard navigation and compatibility.

C) Security Requirements

Kimathi Bus Reservation System must adhere to security and privacy requirements to data and ensure compliance with regulations. They **Data Protection and Privacy:**

The system should comply with data implement measures to safeguard travelers' data, handle access requests, clear privacy External Policies and Regulation: The system must comply with industry regulations and guidelines related to data security Other Requirements

Kimathi Bus Reservation System may have other requirements that are not in the include

Database -The system should use a reliable and scalable database management system to manage traveler's data, bookings and other relevant information. The database should efficient data retrieval and manipulation operations, ensuring optimal system.

Legal - The system must comply with applicable laws and regulations related to data privacy, accessibility, and any other relevant legal obligations in the jurisdiction .

Performance - Specify performance objectives related to response time, system scalability to system performance under varying loads and usage .

Documentation

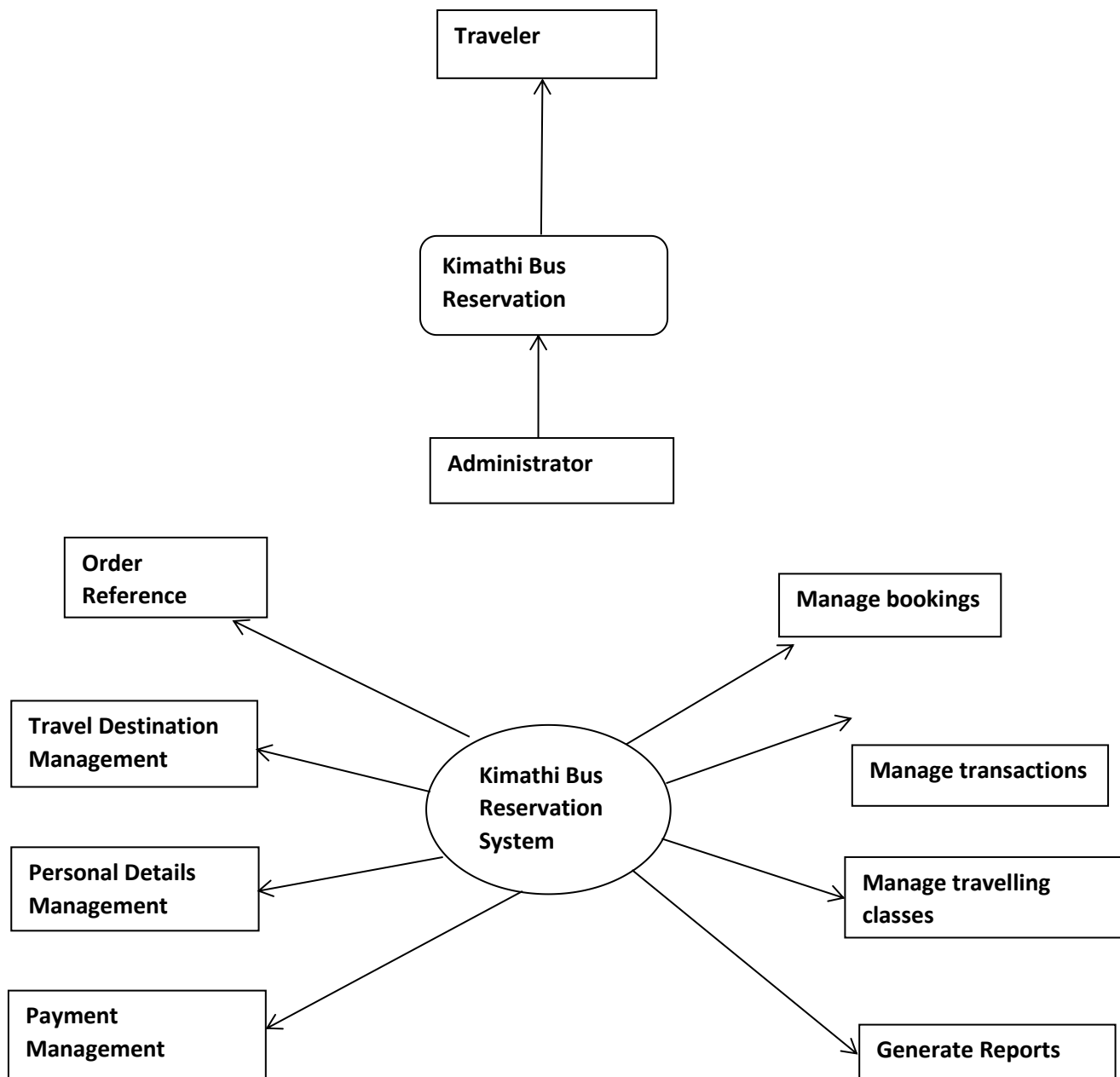
The project should include documentation, such as system architecture, instructions, and user manuals to aid in system understanding, deployment maintenanc

3.0 SOFTWARE DESIGN SPECIFICATION

3.1 System Overview

The Kimathi Bus Reservation System is a modern solution that saves the traveler the hustle of going up-to the stage to find a seat for a bus to their destinations. It allows travelers to book a seat by submitting all the required necessary details including personal info, bank details e.t.c. the system aims to enhance traveler satisfaction and deliver a convenient and efficient service experience.

The following diagram describes an overview of Kimathi Bus Reservation System



3.2 SYSTEM ARCHITECTURE

Kimathi Bus Reservation System will be designed with a layered architecture to ensure modularity, scalability and maintainability. The architecture will incorporate various components that interact with each other to provide a seamless self-service experience for travelers. The following components are essential to the system architecture:

- **Presentation Layer:**

The presentation layer is responsible for the user interface through which travelers interact with the system. It includes a mobile application that provides intuitive and user-friendly interfaces for travelers to book for seats and make payments.

- ii. Application Layer:**

This layer contains the logic and functional components of the system. It handles the processing and coordination of various tasks, such as filling personal details, destination interested in, process payments. This layer ensures the smooth execution of self-service operations and manages flow of data between the presentation layer and the underlying layers.

- iii. Data Layer:**

Comprises the database and data resources required for the system's operation. It stores and manages information, reservation details, payments and other relevant data. Data layer ensures data integrity, security and availability for the system's functionalities.

- iv. Hardware Components:**

Hardware components of the architecture encompass the physical devices that support the system's operation. Servers and network infrastructure are also part of the hardware components, providing the necessary computing and networking resources for system operation.

- V. Network Infrastructure:**

The network infrastructure facilitates communication and data exchange between the different components of the system. It enables seamless connectivity between the mobile devices and servers. Ensures reliable and secure data transmission, facilitating real-time updates and ensuring the system responsiveness.

3.3 Software Design

Kimathi Bus Reservation System is a software solution that allows travelers to book for seats for a given fleet of buses to their preferred destinations.

The proposed system is based on a three-tier client-server architecture design that has a web application module. A user interface is used by the web module to start a new application and for updates. The module is connected to a database, where requests from the clients or the server or any data retrieved. In addition user data can be written to the database. Any interaction between the mobile interface and the database shall be done through a web service. When the client machine sends the data request to the server machine, the request is first received by a middleware then is obtained to the server.

The response is first received by the middle layer which then sends it to the requester, the client machine. All data logic and business logic are stored on the middleware. The use of middleware improves the flexibility of this architecture.

Logical Design

This describes the required functionalities of a system. Mostly concerned with hardware and software requirements and the process to be performed.

Use Case Diagram

This is a diagram that shows the relationship between a set of use cases and the actors involved.

- **Actors**

These are the roles that the users of the system play with respect to the system itself. Actors include: customers (travelers), administrator.

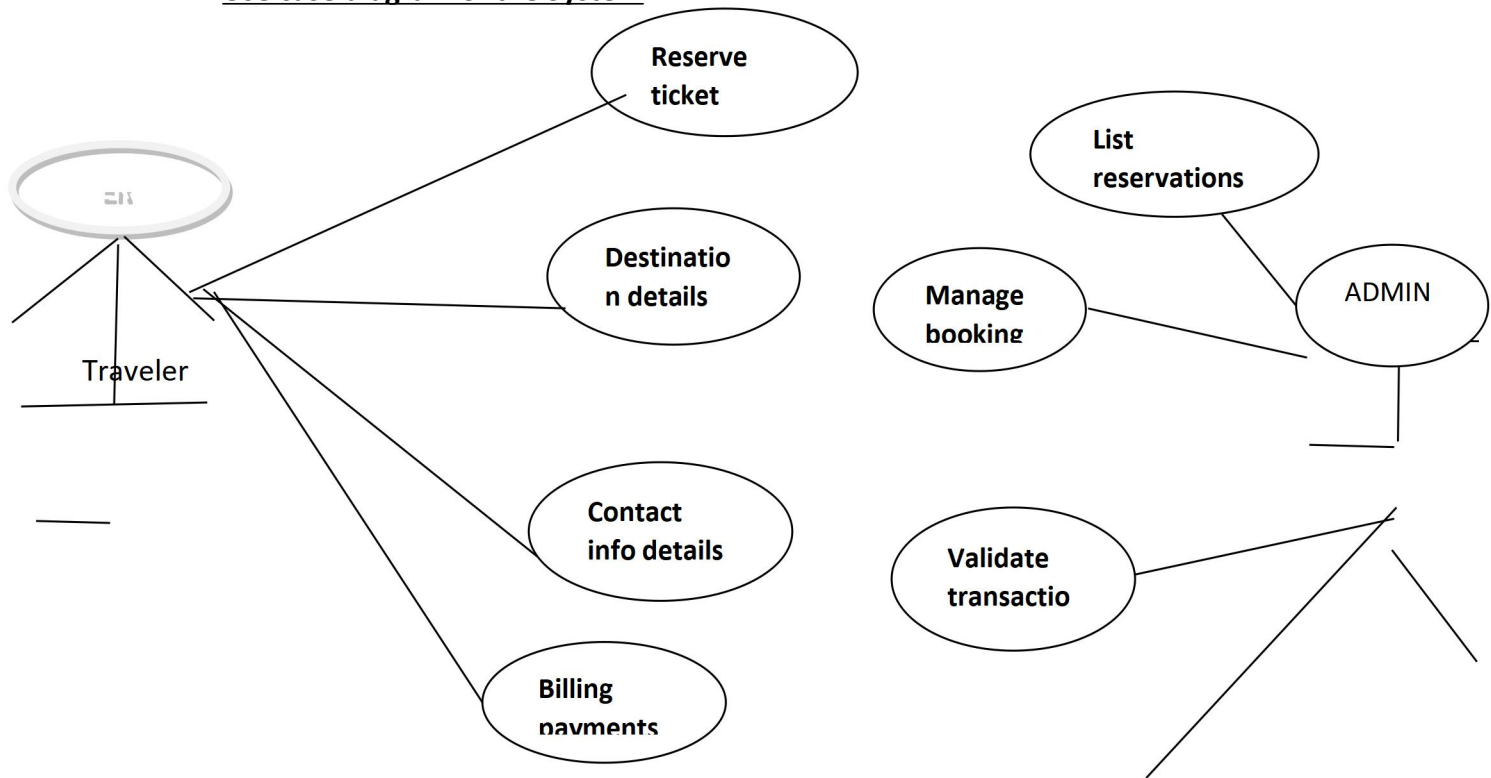
- **Travelers**

They are the main users of the system as they book for their preferred destinations via the website.

- **Administrator**

This a user who is authenticated to have full access to the database and the system at large. He/she oversees management of made bookings, validate transactions and control billings.

Use case diagram of the System



3.3.1 RESERVATION MODULE

Pseudo Code

Start

Generate order reference

Prompt for destination

Prompt for travelling class

Prompt for No of seats, date of travel

Validate submitted details

If validation successful

Confirm Reservation

Prompt for Name

Prompt for Contact

Prompt for Gender

If validation is successful

Proceed to next process

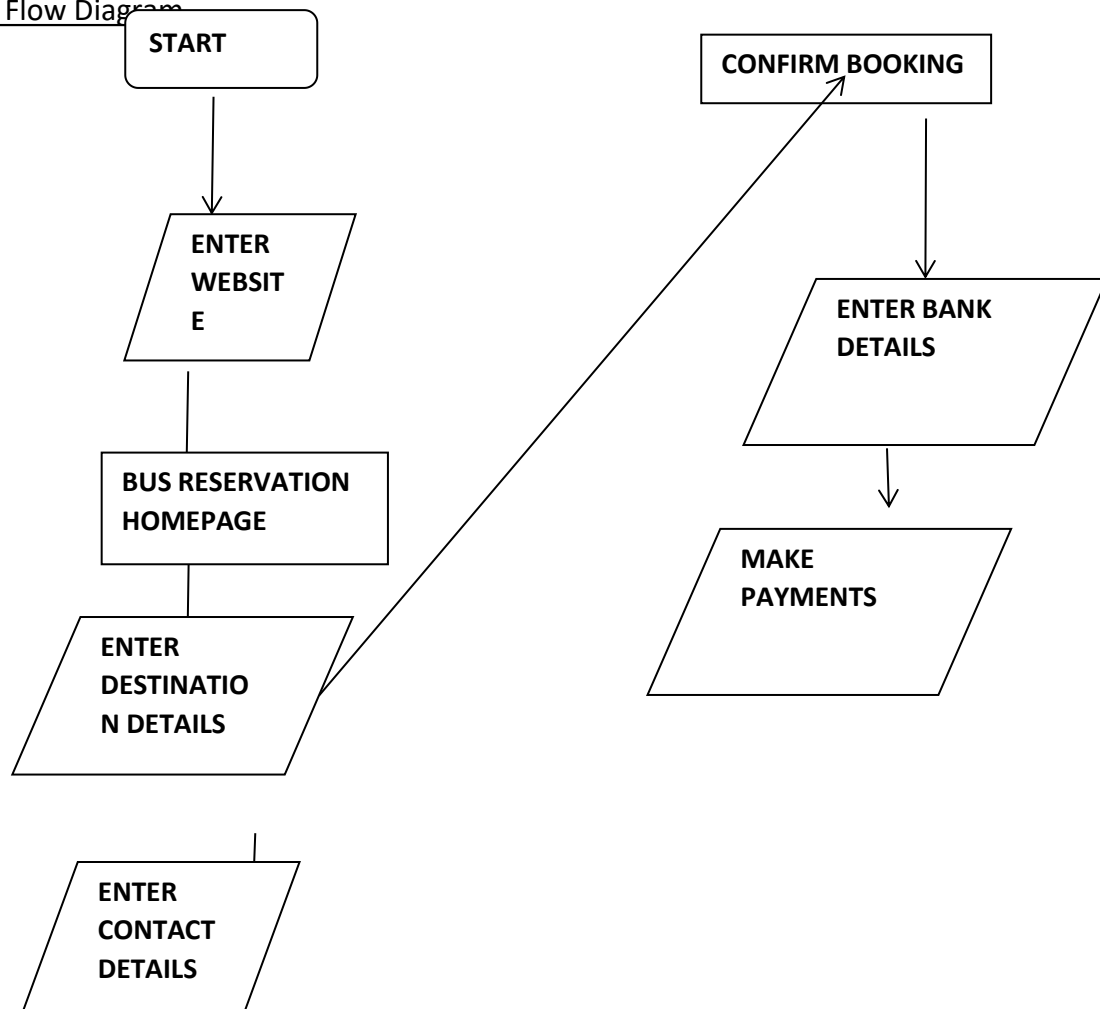
Prompt for Payment

Prompt for Transaction ID

Confirm Payments

End

3.4 Data Flow Diagram



UN NORMALIZED FORM	1 st NORMAL FORM	2 nd NORMAL FORM	3 rd NORMAL FORM
First name	first name	First name	Order ref
last name	Last name	Last name	Last name
Ticket ref	Ticket ref	Ticket ref	transaction

[File and Database Design](#)

The file and database design is useful for the development of the project

[3.5 Database Description](#)

[3.5.1 ADMIN TABLE](#)

<u>SERIAL NO</u>	<u>FIELDS</u>	<u>DATA TYPE</u>	<u>DESCRIPTION</u>
1	Name	Varchar	
2	ID	Int	PK
3	Contact	Int	
4	email	Varchar	
5	username	Varchar	
6	password	Int	

3.5.2 AVAILABLE CLASS TABLE

<u>SERIAL NO</u>	<u>FIELDS</u>	<u>DATA TYPE</u>	<u>DESCRIPTION</u>
1	Class ID	Int	FK
2	Class Name	Varchar	
3	Class Capacity	Int	
4	Class Price	Int	
5	Description	varchar	

3.5.3 BOOKING DETAILS

<u>SERIAL NO.</u>	<u>FIELDS</u>	<u>DATA TYPE</u>	<u>DESCRIPTION</u>
1	Order reference	Varchar	PK
2	Name	Varchar	
3	Contact	Int	
4	Gender	Varchar	
5	Class_reserved	Varchar	
6	Destination	varchar	
7	Seat reserved	Int	
8	Date	date	
9	Transaction ID	Int	FK
10	Account	Varchar	
11	Amount	int	

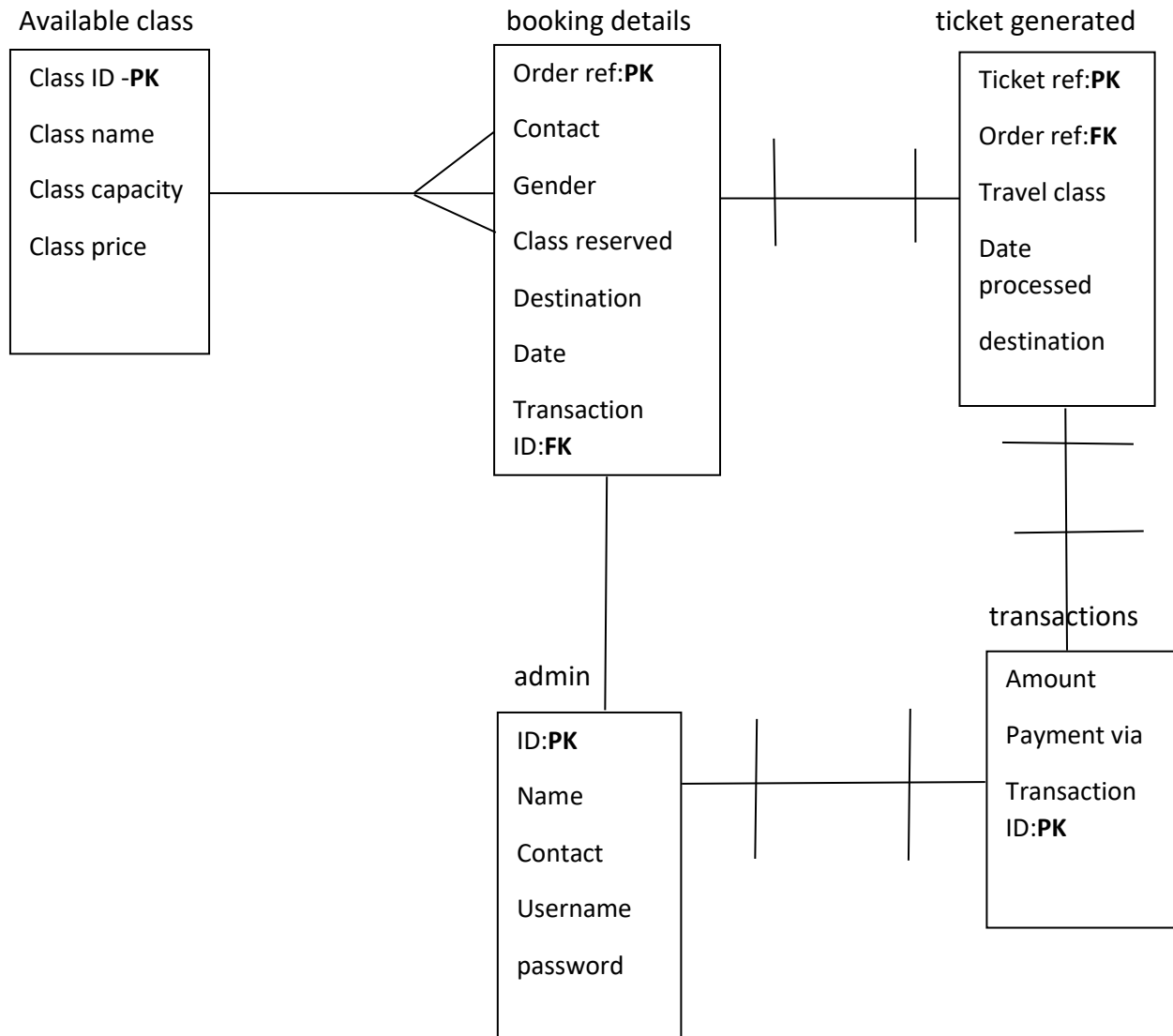
3.5.4 TICKETS GENERATED TABLE

<u>SERIAL NO</u>	<u>FIELDS</u>	<u>DATA TYPE</u>	<u>DESCRIPTION</u>
1	Ticket reference	Varchar	
2	Order reference	Varchar	FK
3	Travel class	Varchar	
4	Date processed	Date	
5	destination	varchar	

3.5.5 TRANSACTIONS TABLE

<u>SERIAL NO</u>	<u>FIELDS</u>	<u>DATA TYPE</u>	<u>DESCRIPTION</u>
1	Amount	Int	
2	Payment via	Varchar	
3	Transaction ID	Int	PK
4	Used	varchar	

3.6 ENTITY RELATIONSHIP DIAGRAM



3.7 System Integrity controls

For a bus reservation system that handles sensitive information, the following should be incorporated to ensure data security, privacy and compliance:

- Internal security: implement robust access controls to restrict access to critical data items based on the principle of least privilege.
- Application audit trails: implement application trails to dynamically record and monitor retrieval access to designated critical data.
- Standard tables: establish standard tables for validating and verifying data fields within the system. This ensures data integrity and consistency by enforcing predefined rules and constraints on data input and updates.
- Verification Processes: define verification processes for any additions, deletions or updates made to critical data. This includes implementing appropriate approval workflows, data validation checks, and change management controls to ensure accuracy and integrity of sensitive information.

4.0 TEST PLAN INTRODUCTION

This section provides an overview of the entire test document, outlining both the test plan and the procedure for the Bus reservation system.

4.1 Goals and objectives

The overall goals and objectives of the test process are as follows:

- To ensure the Kimathi Bus Reservation system functions accurately, efficiently and reliably.
- To verify that the system meets the specified requirements and provide a seamless and satisfactory experience for travelers.
- To identify and rectify any defects to improve the overall system performance and user experience.

4.2 Statement of Scope

The scope of software testing for the Kimathi Bus reservation system includes the following functionality:

1. Reservation and booking management- testing the ability of travelers to make bookings to their preferred destinations, travelling class & no of seats reserved.
2. Contact details management- testing the system's ability to process filled in contact details, full names, phone contact/ e-mail addresses, and gender for a particular traveler.
3. Billing and payment management- testing the ability of the system to process confirmed mode of payments done, together with the transaction codes and amount payable for a given destination.

Exclusions:

- The testing will not cover third-party integration with external systems.
- Hardware-specific testing, such as compatibility with specific devices, will not be included in this test plan.

4.3 Major constraints

The following major constraints will impact the manner in which the software is tested:

- Time constraint: the testing must be completed within the specified project timeline to meet the planned system deploy time.
- Resource limitation: the availability of testing resources, including testing environments and personnel may affect the depth and scope of testing.
- Operational impact: testing activities should not disrupt the normal bus booking operations or negatively affect traveler's experience during the testing process.

4.4 Software to be tested

The software to be tested is the Kimathi Bus Reservation System. Exclusions are noted explicitly, which includes any third-party integration.

4.4.1 Testing strategy

The overall testing strategy for the Kimathi Bus Reservation System is as follows:

A) Unit testing

The strategy for unit testing is as follows:

- Identify software components to undergo unit testing based on their critically and complexity.
- Unit testing will focus on individual units of code to ensure they work as expected in isolation.
- The criteria for selecting components for unit tests will be based on their importance in critical functions and areas prone to errors.

B) Integration testing

The integration testing strategy is as follows:

- Integration testing will verify the interactions between different software functions and components.
- Integration testing will be performed based on the order of integration by software function, ensuring that key functions are integrated and tested first.

C) Validation testing

The validation testing strategy is as follows:

- Validation testing will ensure that the entire system works correctly and meets the specified requirements.
- The order of validation will be conducted by software function, ensuring that each function is validated and its integration with other functions is tested.

D) High- Order testing

The high order testing strategy is as follows:

- High-order testing will cover various aspects such as recovery testing, security testing, stress testing, performance testing, alpha/beta testing.
- Responsibility for each type of high-order test will be assigned to specialized testing resources or teams .

4.5 Testing resources and staffing

The testing resources and staffing for the Kimathi Bus

Reservation system are as follows:

- Specialized testing resources, including testers with expertise in different types of testing (unit testing, integration testing) will be assigned.
- Testing team members will be responsible for different testing phases, ensuring efficient and effective testing.

4.6 Test work products

The work products produced as a consequence of the testing strategy include: -

- Test plans for each testing phase, including unit testing, integration testing, validation testing and high-order testing.
- Test cases and test scripts for each test phase.
- Test reports documenting the test results, defects found and corrective actions taken.

4.7 Test Record Keeping

Mechanisms for storing and evaluating test results will be as follows:

- A centralized test repository will be used to store test cases, test scripts and test reports.
- Test logs will be maintained to track the progress and results of each testing phase.

4.8 Test Metrics

The following test metrics will be used during the testing activity:

- Test coverage metrics to measure the percentage of code tested.
- Defect density to assess the number of defects found per unit code.
- Test execution metrics to measure the time taken to execute each testing phase.

4.9 Testing tools and Environment

The test environment for the Kimathi Bus Reservation System will include:

- Testing tools for test automation, defect tracking and performance testing.
- Simulators to simulate different scenarios and test functionalities.
- Specialized hardware and devices for compatibility and performance testing.

Test Procedure

This describes the detailed test procedure, including test tactics and test cases for the Kimathi Bus Reservation System.

Software to be tested

The software to be tested is the Kimathi Bus Reservation System. Exclusions are noted explicitly which includes any third-party integration.

Unit test Cases

Component:

Generate order reference

Travelling and booking info

Contact info

Payment & verification Verify order details

No stubs or drivers are required for unit testing of this component.

4.9.1 Test Cases for Component

Test case 1: Booking info & travelling

Description: Verify that traveler makes bookings for seats for a vehicle to a particular destination during a given duration.

Test steps:

1. Select the preferred destination.
2. Choose type of travelling class.
3. Reserve No of seats.
4. Set the date of travel.

Expected result: The system should process the booking request and proceed to next page.

Test case 2: Contact Info

Description: Verify that traveler gives correct contact info as required, provide mobile number or e-mail address and gender.

Test steps:

1. Give full name.
2. 2. Fill mobile contact or email-address.
3. Choose gender.

Expected result: The system should process contact information filled and proceed to next unit.

Test case 3: Payment & verification

Description: Verify that traveler gives the correct method of payment with the transaction ID.

Test steps:

1. Choose mode of payment.
2. Give code of the transaction.

Expected result: The system should process the payment made for the class of seats reserved.

Test case 4: Verify order details

Description: Confirm the detail already filled in.

Test steps:

1. Confirm details.

Expected result: The system should confirm details and prints a ticket.

4.9.2 Purpose of Tests for Component

The purpose of conducting the unit tests for component is to ensure the individual functionalities related to travelling & booking details, contact info, payment & verification and verify order details work correctly and independently.

These tests will verify the correctness and the overall functionality of each component.

Expected results for Component

test case 1: The traveler successfully confirms travel details

test case 2: The traveler's contact info is stored.

test case 3: payments & verification are established

test case 4: verification of order details supplied.

Integration Testing

Testing procedure for integration:

Integration testing will involve testing the interactions between different software functions and components related to travelling and booking details, contact info, payment and verification and verify order details.

The integration testing procedure will follow a sequential order to ensure key functions are integrated and tested first.

Stubs and drivers required:

No stubs or drivers are required for integration testing.

Test Cases and their Purpose: -

Integration test cases will be designed to test the interactions and data exchange between the different components related to traveler functionality. The purpose of these test cases is to verify the seamless flow of information and functionality when different components work together.

Expected Results:

The integration test cases should result in successful data exchange and functional interactions between various components, without any system errors or failures.

Validation testing

Testing Procedure for Validation:

Validation testing will involve testing the entire Kimathi Bus Reservation System to ensure it meets all specified requirements and provide appropriate user experience. The validation testing procedure will cover end-to-end user journeys and scenarios.

Test Cases and their Purpose:

Validation test cases will be designed to represent typical user interactions and workflow in the system. The purpose of these test cases is to validate the entire system functions correctly and satisfies user requirements.

Expected results: The expected results for validation test cases should demonstrate that the system works as a whole and provides a satisfactory experience for travelers.

Pass Criterion for all validation tests:

The validation test cases will have specific pass/fail criteria defined based on user requirements and expected system behavior.

High-Order testing (System testing)

High- Order Testing Procedure:

High-Order testing involves conducting various tests to evaluate the overall system performance and behavior.

High-Order Test Methods:

1. Recovery testing: test the system's ability to recover from failures, crashes or unexpected errors.
2. Security testing: Verify the system's security measures to protect traveler data and unauthorized access.
3. Stress testing: Evaluate the system's performance under heavy load or stress condition.
4. Performance testing: Assess the system's overall performance, response time and resource usage.
5. Alpha/beta testing: Conduct real-world testing with a limited number of users to gather feedback and identify potential issues before full release.

Pass/Fail Criteria:

Define the pass/fail criteria for each high-order test method based on the system's requirements and performance expectations.

Specialized testing resources with expertise in unit testing, integration testing, validation testing, high-order testing and system testing will allocated.

Staffing:

The team will consist of qualified testers, test analysts, and a test manager responsible for overseeing the entire testing process.

Test Work Products

- The work products produced as a consequence of the testing procedure include:
- Test plans and test cases for each testing phase
- Test reports documenting the test results, defects found and corrective actions taken

Test record keeping and record

A centralized test repository will be used to store test cases, test scripts and test reports.

Test

Log : A log test will be maintained to keep a chronological record of all tests conducted and their results.

5.0 SYSTEM IMPLEMENTATION

5.1 INTRODUCTION

The Kimathi Bus Reservation System aims to streamline and enhance the traveler experience by providing a user-friendly platform for travelers to manage their bookings of destination travels. This implementation document outlines the plan to develop, install and maintain the proposed system.

PURPOSE OF IMPLEMENTATION

The purpose of implementing the Bus Reservation System is to improve operational efficiency, enhance traveler satisfaction and reduce the tedious work of going to a bus station to book for a bus.

PROPOSED SYSTEM

The proposed system will consist of a web-application that allows travelers to:- book their travel destinations, provide contact information, supply payment details and also verify the order details generated.

5.2 INSTALLATION & CONVERSION PLAN

The installation and conversion plan will consist of the following key phases:

1. Assessment: Evaluate the transport sector of Kimathi Bus Company, technology stack used

- & data security measures to ensure compatibility & compliance with new system.
2. System development: Develop the web-based application on the specified requirements.
 3. Testing: Conduct rigorous testing, including functional usability and security testing, to identify and rectify any issues.
 4. Data migration: Transfer existing traveler's data & booking records to the new system without data loss.
 5. Training: Train administrators to manage the new system at hand.

5.3 IMPLEMENTATION REQUIREMENTS

The implementation requirements include:

- Skilled software developers to build the web-application
- Database administrators to handle data migration and ensure data integrity
- Quality assurance team to conduct thorough testing
- Project managers to oversee the implementation process.
- Trainers to educate travelers and admins on using the system.

TOOLS USED

The following tools will be used in the implementation:

Programming language: PHP

Database: MYSQL

Version Control: Git for code management and collaboration

5.4 USER INTERFACE DESIGN IMPLEMENTATION

1. Generating order reference

Done! Proceed

Admin Login

LOST YOUR TICKET?

Enter Order Ref

Q

Re-print

FAQS

▶ Is there a receipt generated?

▶ Is it 24/7 operational?

2. Travelling and booking info

✈ Booking Details
Travelling and booking info

🏠 Details
Contact information

💰 Billing
Payment and verification

📋 Confirm Details
Verify order details

🖨 Finish and Print
Printing Ticket

Booking Info
Order Ref: **BS64T72R69O** [Cancel Order](#)
Enter traveling booking info

Destination

--Travel Destination--

Traveling Class

--Travel Class--

Number of Seats

1

Date of Travel

ex. August 03, 1998

Submit Details

3. Contact info

Booking Details
Travelling and booking info

Details
Contact information

Billing
Payment and verification

Confirm Details
Verify order details

Finish and Print
Printing Ticket

Enter your Customer Details!
Order Ref: **BS64T72R69O** [Cancel Order](#)
Fill the required Fields

Full name

Full name

Contact/Mobile or Email address

Mobile No./Contact or Email address

Gender

--Choose Gender--

Submit Details

4. [Payment and verification](#)

Booking Details
Travelling and booking info

Details
Contact information

Billing
Payment and verification

Confirm Details
Verify order details

Finish and Print
Printing Ticket

Validate Payment Information
Order Ref: **BS64T72R69O** [Cancel Order](#)
Enter Payment Details to Proceed

Payment

Method of Payment

Transaction ID

Transaction Code






Confirm Amount(Ksh)

1500

Proceed

? Confirm the payment of the High Class number of seats

5. [Confirm details](#)

 Booking Details Travelling and booking info	 Details Contact information	 Billing Payment and verification	 Confirm Details Verify order details	 Finish and Print Printing Ticket
-----------------------------------------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------------	------------------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------

Ticket might not be re-printed, hence details you provided should be valid

THE DETAILS PROVIDED

- TICKET OWNER: james onyango
- DESTINATION: RUIRU to MOYALE
- DATE OF TRAVEL: April 05, 2024
- TRAVEL CLASS: Special Needs Travel
- NUMBER OF SEATS: 1
- AMOUNT PAYING: 1500 Via KCB_BANK Transaction ID: 01222247789

CONFIRM DETAILS

[YES|Confirm](#)

6. [Admin Access Panel](#)

Admin Access Panel!

Username

admin

Password

☐ Remember me

Login!

5.5 [OUTPUT OF THE SYSTEM](#)

The expected outputs of the Kimathi Bus Reservation System include:

- Improved bus booking reservation management
- Efficient handling of traveler's info
- Enhanced traveler engagement through personal info

HARDWARE AND SOFTWARE NEEDED

The hardware and software needed to run the Kimathi Bus Reservation System are as follows:

HARDWARE

Web servers to host the web application

Network infrastructure to support data communication

SOFTWARE

- Operating System (Windows) for web servers and database servers
- Web Server Software (XAMPP) for hosting the web application
- Database management System (e.g., MySQL) for data storage
- Security Software to protect the system from potential threats

SOFTWARE MAINTENANCE PLAN

The software maintenance plan includes:

- Regular updates to fix bugs and security vulnerabilities
- Monitoring and performance optimization to ensure system stability

- Backup and data recovery procedures to prevent data loss
- Support for new operating system versions and devices.
- Periodic reviews and enhancements based on user feedback

5.6 TRAINING OF USERS

The training of users will involve:

- Conducting training sessions for administrators on using the system to manage traveler information.
- Providing online tutorials and guides for travelers to familiarize themselves with the self service features.
- Offering a help desk or support team to address any questions or difficulties faced by users during their interactions with the system.

SYSTEM LIMITATIONS

Despite the successful testing outcomes, a few limitations were identified during testing phase:

1. **Internet Connectivity:** The system requires a stable internet connection for real time updates and traveler's interactions. Offline capabilities should be considered for areas with limited connectivity.
2. **Device Compatibility:** The system was primarily tested on standard devices and compatibility issues may arise on certain older devices or less common platforms.

RECOMMENDATIONS

So as to further enhance the Kimathi Bus Reservation System, the following recommendations are proposed:

1. **Offline Mode:** Implement an offline mode feature to allow travelers to perform essential tasks without an active internet connection.
2. **Device Testing:** Conduct extensive testing on a wide range of devices to ensure optimal performance and compatibility across various platforms.
3. **Regular Security audits:** Conduct regular security audits to identify and address any potential vulnerabilities, ensuring the protection of traveler's data.
4. **Continuous Improvement:** Encourage feedback from travelers to continuously improve the system's functionality and user experience.

5.7 REFERENCES

1. Melisa, K. (2007). Online Bus ticketing system: University of Malaykuala Lumpur Accessed 17th November 2014.
2. Mezghani, M . (2008). Study on electronic ticketing in public transport. Available at: <http://www.emta.com/IMG/pdf/EMTA-ticketing.pdf> accessed: 16th November 2014.
3. Badre, A (2002): shaping web usability. Boston: Pearson Education, Inc.

•