KIGALI INDEPENDENT UNIVERSITY ULK

SCHOOL OF SCIENCE AND TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE

P.o Box: 2280 KIGALI

Website: https://ulk.ac.rw/

ONLINE BOOKING WEDDING VENUE MANAGEMENT SYSTEM

Case study: AmaliTech

Prepared by: Magnifique UMURERWA

Roll Number: 202111058

Supervisor: Mr. RUTARINDWA Jean Pierre

Dissertation Submitted in the School of Science and Technology in Partial Fulfillment

Of the requirements for the award of Bachelor's Degree in Computer Science.

September, 2024

Declaration

I, Magnifique UMURERWA, hereby declare that this work entitled "Online Booking Wedding Venue Management System "Submitted in partial fulfilment of the requirement for the award of Bachelor's degree in computer science, is our original work and has not been presented for other University

Student Name Magnifique UMURERWA

Date.....

Signature.....

Approval

This dissertation entitled “ONLINE BOOKING WEDDING VENUE MANAGEMENT SYSTEM’’ has been done under my supervision and submitted for examination with my approval.

Supervisor Name: Mr. RUTARINDWA Jean Pierre

Date: ……/……/…… .

Signature: …….. .

Dedication

With Genuine Gratitude,

I dedicate this Research Project

To my mother and my brother,

To the family of Joseph NGOMIJANA,

To my aunt Pauline MUSHIKIWE,

To my friend Mosey

To all my friends and relatives,

To my supervisor,

To All Lecturers and my colleagues at ULK.

Acknowledgement

First and foremost, I would like to acknowledge and thank God for providing me with the strength, wisdom, and guidance throughout this research journey.

I am deeply grateful to the founder and President Prof.Dr. Rwigamba Balinda of ULK (*Kigali Independent University*) for establishing such a prestigious institution that has provided me with the platform to pursue my academic aspirations.

I would like to extend my heartfelt appreciation to my supervisor for his valuable guidance, constructive feedback, and unwavering support. His expertise and insights have played a pivotal role in shaping the direction and quality of this research.

I would also like to express my gratitude to the respondents who participated in this study. Their willingness to share their experiences and insights has been instrumental in generating meaningful findings and conclusions.

Furthermore, I would like to acknowledge the academic staff at ULK for their dedication to excellence in education. Their commitment to fostering a conducive learning environment has greatly contributed to my personal and intellectual growth.

I am also indebted to the administrative staff at ULK for their efficiency and assistance throughout the administrative processes involved in this research.

I would like to extend my appreciation to my family members for their unwavering support, understanding, and encouragement throughout this research endeavor. Their love and belief in me have been a constant source of motivation.

Lastly, I would like to express my gratitude to my friends who have been there for me, providing encouragement, and offering valuable insights during this research journey. I am genuinely thankful to all the individuals and institutions mentioned above for their contributions and support, without which this research would not have been possible.

Magnifique UMURERWA

Table of contents

List of Figures

List of abbreviations and acronyms

ACID: Atomicity Consistency Isolation Durability

API: Application Programming Interface

CSS: Cascading Style Sheet

DBMS: Database Management Systems

DDD: domain-driven design

HTML: Hypertext Markup Language

JWT : JSON Web Token

MDA: Model-driven architecture

ODM: Object Data Modeling

OOM : object-oriented methodology

OOSAD: Object-Oriented System A nalysis and Design Methodology

SDLC: Software Development Life Cycle

SQL: structured query language

SRS: Software requirements specifications

UML: Unified Modeling Language

URI: Uniform Resource Identifier

WCAG: Web Content Accessibility Guidelines

Abstract

Planning a wedding can be both thrilling and challenging, particularly when it comes to finding and booking the ideal venue. In Kigali, Rwanda, couples face difficulties due to the absence of a comprehensive online resource for wedding venues. This lack of centralized information complicates the comparison of options based on factors such as location, size, aesthetics, and cost, while the traditional booking process is often marred by cumbersome phone calls, in-person visits, and complicated paperwork.

To alleviate these issues, this project introduces the Online Booking Wedding Venue Management System, designed to streamline the venue selection and booking process for couples in Kigali. The system provides a centralized database of available wedding venues, accessible through a user-friendly web interface. Built using HTML, CSS, Node.js, Express, and MongoDB, the platform simplifies the search for and reservation of wedding venues, thereby reducing stress and saving time for users.

The project aims to address the challenges faced by couples by offering a seamless and efficient booking experience. The system's core features include real-time availability tracking, detailed venue descriptions, and an intuitive booking process. By improving accessibility to venue information and facilitating a more straightforward booking procedure, the platform provides significant benefits to couples, venue managers, and related stakeholders.

In summary, the Online Booking Wedding Venue Management System addresses key issues in the wedding planning process in Kigali by offering an innovative, user-centric solution. The system enhances convenience, reduces administrative burdens, and improves overall user satisfaction. Future enhancements could include mobile application development and advanced features to further refine user experience and operational efficiency.

Keywords: Online, Booking, Weeding, Management, system

CHAPI: GENERAL INTRODUCTION

1.0. INTRODUCTION

Planning a wedding is an exciting but often stressful experience, especially when it comes to finding the perfect venue. In Kigali, Rwanda, couples face significant challenges due to the lack of comprehensive online information about wedding venues. This makes it difficult to compare options based on location, size, aesthetics, and cost. The booking process is also cumbersome, involving numerous phone calls, in-person visits, and confusing paperwork, which adds unnecessary stress to an already demanding task.

To address these issues, this project introduces the ONLINE BOOKING WEDDING VENUE MANAGEMENT SYSTEM. This platform aims to simplify the process of finding and booking wedding venues for couples in Kigali by providing a centralized database of venue options and a user-friendly booking interface. By streamlining the search and booking processes, the system saves time and reduces stress for couples.

This chapter provides an overview of the research project, outlining the motivation, problem statement, and objectives. It describes the challenges couples face when booking wedding venues in Kigali and explains how the new system aims to address these issues. The chapter also details the scope of the study, the research questions, and the methodology used to develop the platform. Lastly, it highlights the potential benefits of the project for various stakeholders, including couples, venue managers, and institutions. By understanding these elements, we can appreciate the significance and potential impact of the ONLINE BOOKING WEDDING VENUE MANAGEMENT SYSTEM.

1.1 Background of the Study

The tourism industry includes a lot of different parts, like events. Events have become really important lately, both in real life and in studying them. Think about throughout history, events have always been a big deal, whether it's marking something important or just celebrating someone's special moment. Even today, things like weddings are super important to people all over the world.

Weddings, especially, are huge milestones. They're about two people coming together and starting a new chapter in their lives. Whether it's a big fancy affair or just a small gathering with close friends and family, weddings are seen as one of the most significant moments in someone's life. They're not just about the party; they're about the emotions and memories that come with it. As Davis (2000) puts it, "a wedding is the celebration of a lifetime, regardless of whether it takes the form of a grand spectacle for hundreds of guests or an intimate gathering of loved ones."

Choosing the right place for a wedding is crucial. The venue sets the stage for the whole event, creating the atmosphere and vibe that everyone will remember. It's usually one of the first things couples decide on because it affects so much else about the wedding (Daniels et al., 2012).

So, understanding how happy people are with the wedding venue is really important. It's about looking at how the venue adds to the whole experience and how people feel about it afterwards compared to what they expected. That's key to making sure guests have a great time and leave with happy memories.

1.2 Problem Statement

Planning a wedding is a complex and stressful process, and finding the perfect venue is one of the most important decisions couples will make. However, the current process of searching for and booking a wedding venue in Kigali is inefficient and time-consuming. Couples often spend hours making phone calls and visiting venues in person, only to find that the information they need is incomplete or outdated. This adds unnecessary stress to an already demanding time.

Specifically, in the city of Kigali, there are many places that can accommodate wedding ceremonies, but there are no automated or computerized systems to help people easily find information about available venues and the services they offer. This leads to several problems:

Challenges for Couples:

- Lack of centralized online information about wedding venues in Kigali.
- Difficulty finding venues that meet their specific needs and budget.
- Time-consuming and frustrating process of contacting venues individually.
- Limited access to virtual tours or detailed venue descriptions.

- Uncertainty about real-time availability.

Challenges for Venue Managers:

- Manual booking processes are inefficient and prone to errors.
- Difficulty managing inquiries and communication with multiple couples.
- Limited marketing reach and inability to showcase venues effectively online.
- Challenges in managing deposits and payments securely.

To address these problems, the research aims to develop the ONLINE BOOKING WEDDING VENUE MANAGEMENT SYSTEM. This platform will simplify the process for both couples and venue managers by providing a centralized database of Kigali wedding venues. It will allow couples to search and compare options based on their needs and streamline the booking process with features like real-time availability information and secure online payments.

1.3Research objectives

1.3.1 General objectives

To develop and implement an ONLINE BOOKING WEDDING VENUE MANAGEMENT SYSTEM that simplifies the process of finding and booking wedding venues for couples in Kigali, Rwanda.

1.3.2Specific objective

- To assess the current system of booking wedding venues in Kigali.
- To provide a centralized and comprehensive database of wedding venues in Kigali.
- To develop a user-friendly and reliable interface for online booking wedding venues platform.
- To streamline the booking process and provide real-time availability information.
- To develop a system from which reports can be generated.
- To develop system that will measure security.

1.4Research questions

- What are the challenges that couples face when booking wedding venues in Kigali?

- What features would be most important to couples in an online booking wedding venues system?
- How can an online booking wedding venues system be designed to meet the needs of both couples and venue owners?
- How can the booking process be streamlined and real-time availability information be provided?
- How can an online booking wedding venues system be implemented and tested?

1.5 Scope of the study

The scope of this project involves the development and implementation of an online booking wedding venue management system. The objective is to create a digital platform that facilitates the booking process for wedding venues, enhances transparency, improves data management, fosters collaboration among stakeholders, and ensures efficient management of wedding venue reservations.

1.5.1 Content Scope

The online booking wedding venue management system encompasses various features to ensure a seamless and transparent booking process. The content scope of the system includes:

User Registration and Authentication: The system will enable users, including couples, venue managers, and administrators, to register and create authenticated accounts securely.

Venue Booking Process: A user-friendly interface will facilitate the booking process, allowing couples to select their desired wedding venue, specify date and time preferences, and submit booking requests.

Availability Management: Venue managers can update and manage the availability of their venues through the system, ensuring accurate and up-to-date information for couples.

Booking Confirmation and Payment: Upon submission of a booking request, couples will receive instant confirmation and be prompted to complete payment securely through the system.

Communication and Collaboration: The system will facilitate communication between couples and venue managers, allowing for inquiries, customization requests, and coordination of wedding details.

Reporting and Analytics: Administrators can generate reports and analyze booking trends to optimize venue utilization, marketing strategies, and overall operational efficiency.

1.5.2 Geographical Scope

The researcher focused on wedding venues within the city of Kigali, Rwanda, located in the central-east part of Africa. This project aims to streamline the wedding venue booking process specifically for couples and venue owners within the Kigali metropolitan area.

By focusing on Kigali as the primary geographical scope, the system aims to enhance the wedding planning experience for couples and venue owners within this dynamic urban center.

1.5.3 Time Scope

The study aims to utilize data spanning the past ten years (2014-2024) to inform the development of the online booking wedding venue management system. However, there might be limitations in acquiring reliable data, particularly for the earlier years (2014-2019). Wedding venue booking systems are a relatively recent development, and data from that period might be scarce or inconsistent.

1.6 Project Methodology

Project methodology refers to a systematic approach or framework that outlines the processes, activities, and guidelines to be followed during the execution of a project. Here, we describe the data collection techniques, software development methodology, and system analysis and design method used in the development of an online booking wedding venue management system:

1.6.1 Data collection tools.

For data collection, researcher relied on thorough documentation from various sources and conducted insightful interviews with potential users and venue owners, ensuring a comprehensive understanding for the project.

Documentation

Documentation was a critical aspect of my research process. I utilized various techniques to gather relevant information. These included consulting books, reviews, class notes, and web pages specifically related to online booking wedding venue management systems.

Interview

Additionally, researcher conducted interviews to gain deeper insights into the requirements within the domain. Informal interviews with potential users and venue owners helped me understand their needs and preferences better.

1.6.2 Software Development Methodology

For software development, researcher embraced Agile Software Development methodologies, emphasizing iterative development, collaboration, and continuous improvement. This approach allowed for regular reviews and adjustments to evolving requirements, ensuring a responsive and adaptable development process tailored to the dynamic nature of wedding planning.

1.6.3 System Analysis and Design Method

My approach to system analysis and design centered around Object-Oriented System A nalysis and Design Methodology (OOSAD). This approach models the system as a collection of interacting objects, each representing an essential entity within the system. Use case diagrams were used to define interactions between users and the system, while class diagrams represented data entities and their relationships. Sequence diagrams helped illustrate the flow of events during specific processes, providing a comprehensive understanding of system behavior and structure.

1.7 Significance of the Project

The online booking wedding venue management system holds significant importance and generates interest due to several reasons:

1.7.1 Personal Interest

For those involved in developing an online booking wedding venue management system, there are numerous valuable gains:

Financial Rewards: I can benefit financially through project contracts, potential employment opportunities, or by commercializing the system.

Skill Enrichment: The development process enhances technical skills in web development, databases, security protocols, and system integration.

Industry Contribution: Understanding the intricacies of wedding venue management allows for effective contributions to the event management and hospitality sectors.

Portfolio Enhancement: The completed project serves as a powerful addition to one's portfolio, showcasing the ability to handle intricate projects and attracting potential clients or employers.

1.7.2 Institutional Interest

ULK Institutions: When a university's student develops an ONLINE BOOKING WEDDING VENUE MANAGEMENT SYSTEM, the university gains enhanced reputation, industry engagement, research contributions, student motivation, and a tangible showcase of practical skills. This achievement fosters connections with employers, alumni, and industries, while also showcasing the institution's relevance and innovation to the broader community.

1.7.3 Public Interest

The implementation of an online booking wedding venue management system serves the following public interests, benefiting various stakeholders:

Couples and Businesses: Couples and businesses benefit from a streamlined and efficient venue booking process, with easy access to information, clear guidelines, and standardized forms, simplifying the process of finding and booking wedding venues.

Support for Local Businesses: By facilitating connections between couples and local venues, the system can boost the local economy and support small businesses.

1.8. Limitations of the Project

The Online Wedding Venue Management System for Kigali City has specific limitations for its three main users: clients, vendors, and managers. Clients may face limited payment options and potential booking conflicts due to delayed updates from vendors, as real-time availability tracking is not always guaranteed. Vendors have restricted control over their listings, with limited customization options for venue details and pricing. Additionally, they rely on timely updates to avoid double bookings. Managers may encounter challenges in efficiently managing the system,

as reporting tools and administrative features are basic, requiring manual intervention for tasks such as venue approvals, issue resolution, and generating detailed reports.

1.9. Arrangement of the project

This study is arranged into four chapters:

Chapter 1. General introduction: Introduces project motivation, objectives, and scope. It outlines the problem statement and methodology.

Chapter 2. Literature review: Defines key system concepts, offering a comprehensive literature review. It explores the system environment and techniques.

Chapter 3. System Analysis and Design: Delves into logical system concepts, detailing UML representations. It includes use case, sequence, and activity diagrams.

Chapter 4. System Implementation: Showcases technical system implementation. It discusses software requirements, development processes, and deployment strategies.

CHAPTER II: LITERATURE REVIEW

2.0. Introduction

A literature review is an essential component of academic research that provides a comprehensive overview and critical evaluation of existing scholarly works, theories, and findings relevant to a specific research topic or question. It serves as a foundation for the research and helps to establish the context, significance, and gaps in knowledge that the current study aims to address (Booth, Sutton, & Papaioannou, 2016). The primary objective of a literature review is to demonstrate the researcher's understanding of the existing body of knowledge and to identify the most important and pertinent studies related to their research area (Hart, 2018). By examining a wide range of published works, including journal articles, books, conference papers, and other academic sources, the literature review aims to synthesize and analyze the findings, methodologies, and theoretical frameworks of previous research (Ridley, 2012).

2.1 Definition of Key Terms

In the context of this project, the following key terms related to online booking wedding venues management system are defined:

2.1.1 Online Booking Weeding Venues Management System

An Online Booking Wedding Venues Management System is a sophisticated digital platform designed to streamline the process of booking and managing wedding venues. It combines the functionalities of venue booking engines with integrated venue management capabilities, aiming to enhance the efficiency and effectiveness of event planning and execution (Tonisha Parra, 2023).

2.1.2 Weeding Venues

A Wedding Venue refers to any location where weddings can be held. It encompasses a wide variety of settings, ranging from traditional halls and banquet rooms to more unconventional spaces like gardens, farms, barns, and even country clubs. Each venue offers a unique atmosphere and set of amenities that cater to different tastes and preferences of couples planning their weddings (Collins Dictionary).

2.1.3 Management System

In the context of this project, a management system refers to an integrated platform designed to facilitate and streamline the process of booking wedding venues. This system incorporates several functionalities aimed at enhancing the user experience for both couples seeking venues and venue managers (Maier, R., & Hadrich, T.,2011).

2.1.4 Online Booking

Online Booking refers to the process of reserving or scheduling services, appointments, or events through digital platforms. This method allows customers to select their preferred time slots, make payments, and receive automatic booking updates, all from the comfort of their homes or on the go (Alzua-Sorzabal A, Gerrikagoitia JK, Torres-Manzanera E (2013)).

2.1.5 Communication and Collaboration

Communication and Collaboration refer to the features and tools integrated into the online booking wedding venue management system that facilitate effective interaction and coordination between couples and venue managers. These features are designed to ensure that all parties involved can easily communicate, share information, and collaborate on planning the wedding (Poling, C., 2015).

2.2 Other Related Literatures

2.2.1 Database Literature

Database Management Systems (DBMS):

The evolution of database management systems has significantly impacted the efficiency of data storage and retrieval processes. Traditional relational databases, such as MySQL and PostgreSQL, have been widely adopted due to their robust ACID (Atomicity, Consistency, Isolation, Durability) properties and structured query language (SQL) capabilities. However, with the increasing complexity of modern applications, NoSQL databases like MongoDB and Cassandra have gained prominence for their flexibility and scalability, particularly in handling unstructured data (Stonebraker & Hellerstein, 2005).

Data Modeling Techniques:

Effective data modeling is crucial for designing a database that accurately represents real-world entities and relationships. Modern approaches to conceptual and ER modeling allow designers to create detailed diagrams that capture complex scenarios, adapting traditional methods for current database technologies (Elmasri & Navathe, 2017). Additionally, relational schema design principles, including normalization and denormalization, continue to evolve to optimize database performance and ensure data integrity, particularly in large-scale systems (Silberschatz, Korth, & Sudarshan, 2020). With the rise of NoSQL databases, new data modeling challenges have emerged, requiring adaptations to traditional relational methods to handle distributed, schema-less environments (Sadalage & Fowler, 2013).

Database Optimization Methods:

Performance optimization is a critical aspect of database management. Techniques such as indexing, query optimization, and transaction management play a vital role in enhancing the speed and efficiency of database operations. Research by Garcia-Molina et al. (2008) has demonstrated various indexing strategies that significantly reduce query response times, while transaction management ensures data consistency and concurrency control.

2.2.2 Design Literature

User Interface (UI) and User Experience (UX) Design:

Creating an intuitive and accessible user interface is essential for the success of any software application. Design principles outlined by Norman (2013) emphasize the importance of user-centered design, focusing on usability, consistency, and feedback mechanisms. Accessibility guidelines, such as the Web Content Accessibility Guidelines (WCAG), ensure that applications are usable by people with diverse abilities (W3C, 2018).

System Architecture Frameworks:

The choice of system architecture significantly influences the scalability, maintainability, and performance of an application. The client-server architecture, which separates the client and server responsibilities, has been a traditional approach for many web applications. However,

microservices architecture, which decomposes the application into smaller, loosely coupled services, has gained popularity for its scalability and flexibility (Newman, 2015).

Software Engineering Methodologies:

Object-oriented design (OOD) provides a structured approach to software development by encapsulating data and behavior into objects. This methodology enhances code reusability and maintainability (Booch, 1994). Model-driven architecture (MDA) and domain-driven design (DDD) offer advanced frameworks for managing complex software projects. MDA focuses on creating abstract models that can be automatically transformed into code, while DDD emphasizes the importance of aligning the software design with the business domain (Evans, 2003).

II.3. Discussion on other related systems

The Knot is a popular wedding planning platform from US, that offers a vendor directory that includes venues across the US. It allows couples to search by location, browse venue photos and descriptions, and sometimes even contact venues directly through the platform.

WeddingWire (US): Similar to The Knot, WeddingWire operates within the US and offers a vendor directory with venue listings. It allows couples to search by location, size, and budget. They can view venue details, photos, and potentially contact venues for inquiries.

Easy Weddings (Australia): This website caters to the Australian wedding market and offers a comprehensive online wedding planning resource. It includes a venue directory with search filters, detailed listings, photos, and sometimes even virtual tours.

Let’s consider WeddingWire from US (Timothy Chi, 2005)

2.2.1 WeddingWire

WeddingWire is a leading technology company that connects engaged couples with event professionals. It offers tools and services for merchants, including advertising, marketing, and CRM solutions. With a database of over 200,000 event professionals and more than 2 million consumer reviews, WeddingWire helps couples search, compare, and book vendors (WeddingWire, n.d.).

2.2.2 The purpose of WeddingWire

The purpose of WeddingWire is to connect engaged couples with wedding vendors, specifically focusing on venues but also including other wedding professionals. Here's a breakdown of its key functionalities:

- Venue Directory: WeddingWire offers a comprehensive directory of wedding venues across various locations (primarily focused on the US market).
- Search and Filtering: Couples can search for venues based on factors like location, size, budget, style (e.g., rustic, elegant), and desired amenities.
- Detailed Venue Listings: Each venue listing typically includes information like photos, descriptions, pricing (may vary), and sometimes even virtual tours.
- Communication Tools: WeddingWire may offer features for couples to contact venues directly through the platform to inquire about availability or request quotes.
- Vendor Marketplace: Beyond venues, WeddingWire connects couples with other wedding vendors such as photographers, caterers, florists, and more. This provides a one-stop shop for couples to research and potentially contact multiple vendors for their wedding needs.
- Inspiration and Planning Tools: The platform may offer additional resources like wedding blogs, inspiration galleries, and planning tools (e.g., budgeting checklists) to assist couples throughout the wedding planning process.

2.2.3 Positive Aspects

- Extensive Network: WeddingWire boasts a vast network of vendors, providing couples with a wide array of options for their wedding planning needs. This extensive selection can be particularly beneficial for couples looking for specific services or styles (Chelsea Chaffin, 2020).
- Customer Engagement: Some users praise the platform for its ability to engage with potential clients, highlighting the value of having a presence on WeddingWire for vendors. The platform's visibility can attract attention from couples planning their weddings (Paul Heeren, 2022).
- Support for Small Businesses: WeddingWire positions itself as a supporter of small businesses, offering a platform for vendors to showcase their services and connect with

couples. This can be seen as a positive aspect for vendors looking to grow their business (Fred Davoli, 2023).

2.2.4 Negative Aspects

According to the WeddingWire Reviews these are negative aspects:

- Quality of Leads: Many vendors express frustration with the quality of leads generated through WeddingWire, stating that a significant proportion of leads are either fake or unresponsive. This raises concerns about the return on investment for vendors investing in advertising on the platform (Roxi Montgomery, 2021).
- Customer Service Issues: Complaints about customer service include difficulties in canceling contracts, unexpected charges, and a perceived lack of transparency in the renewal process. These issues suggest potential problems with the platform's business model and its treatment of vendors (Jaime Glez, 2022).
- Misleading Practices: Allegations of misleading practices, such as deleting negative feedback and creating an illusion of a busy platform through fake vendor accounts, have been raised. This undermines the credibility of the platform and its commitment to fairness and transparency (Laura Mary, 2023).
- Advertising Effectiveness: Concerns about the effectiveness of advertising on WeddingWire have been voiced, with claims that the platform fails to deliver on its promise of generating high-quality leads. This skepticism affects vendors' willingness to invest in advertising on the platform (Neha Mehrotra, 2020).

CHAPTER III: SYSTEM ANALYSIS AND DESIGN

3.0 Introduction

Requirements Analysis is the process of defining user expectations for a new software being built or modified. It includes all the work done to determine the needs of different stakeholders (Sommerville, 2011). This process involves collecting and analyzing the services that the system offers (Pressman, 2014). A detailed investigation of something's components or structure is referred to as an analysis, meaning dissecting something into its constituent parts and then coming to your own conclusions by asking yourself WHY and HOW questions (Bohem, 1981).

Performing analysis involves reviewing the current system and recording its specifications. System analysis aids analysts in comprehending the operation and purpose of the current system (Kendall & Kendall, 2011). Software Requirements Specifications (SRS) documents for the proposed system are created at the conclusion of the analysis when a deliverable document is required (Sommerville, 2011).

Examining, purifying, manipulating, and modeling data is the process of data analysis, which aims to unearth relevant information, inform conclusions, and assist decision-making (Chen, Chiang, & Storey, 2012). Data analysis has many features and methodologies in various fields, incorporating different techniques under a variety of titles (Witten, Frank, & Hall, 2016).

A specific method of data analysis known as "data mining" focuses on modeling and knowledge acquisition for predictive rather than merely descriptive goals (Han, Kamber, & Pei, 2012). In this third chapter, we will analyze the existing system by identifying problems that users face and design the proposed system to address those problems using system analysis and design tools and techniques (Sommerville, 2011; Pressman, 2014).

3.1 Analysis of the current system

The current system for booking wedding venues relies heavily on manual processes and traditional methods. Couples interested in booking a wedding venue typically have to visit various venues in person. Prospective clients must physically travel to different locations to assess their suitability, which often involves scheduling appointments and making multiple trips, proving to be time-

consuming and inconvenient. Additionally, couples frequently need to contact venues via phone or email to inquire about availability, pricing, and other essential details.

3.1.1 Problem of the current system

The reliance on manual processes and traditional methods in the current system presents several challenges:

Inefficiency: The need to physically visit multiple venues and communicate through phone or email results in a time-consuming and inefficient process. This can be particularly burdensome for couples with limited time or those planning their wedding from a distance.

Lack of Centralized Information: There is no centralized platform where couples can access comprehensive and up-to-date information about wedding venues. This leads to fragmented and inconsistent data, making it difficult for couples to compare options effectively.

Limited Accessibility: The current system may not be easily accessible to all couples, particularly those with mobility issues or those living far from the venue locations. This limits their ability to explore all available options.

Inaccurate Availability: Manual tracking of venue availability can lead to errors and miscommunication. Couples may find that a venue they are interested in is already booked for their desired date, leading to frustration and the need to restart the search process.

Inconsistent Communication: Relying on phone calls and emails for communication can result in delays, missed messages, and misunderstandings. This can complicate the planning process and lead to misaligned expectations between couples and venue managers.

Stress and Frustration: The inefficiencies and challenges associated with the current system can add to the stress and frustration of wedding planning, detracting from the overall experience.

3.3 Analysis of the new system

3.3.1 Introduction

The new online booking wedding venue management system aims to revolutionize the current venue booking process by providing a comprehensive, centralized digital platform. This system

will allow couples to browse, compare, and book wedding venues effortlessly from the comfort of their homes. By integrating real-time availability, detailed venue descriptions, virtual tours, and secure payment options, the platform will streamline communication between couples and venue managers, enhancing efficiency and satisfaction for both parties. This innovative approach addresses the limitations of the current manual processes, offering a more convenient, transparent, and user-friendly experience.

3.3.2 System requirements

The features of the Online Booking Wedding Venues management system include functional and non-functional requirements. Functional requirements are those requirements that are easier to be found at the beginning while Non-functional requirements are those that define the system properties and constraints.

3.3.2.1 Functional requirements

- The system must allow users to search for wedding venues in Kigali.
- The system must allow users to book a wedding venue and receive a confirmation email
- The system will have a user authorization process that requires users to log in using a username and password.
- The system will allow venue's owner to register their venues
- The system will allow venue's owner to manage their venues
- The system will allow to generate reports
- The system will allow venue's owner to manage their clients

3.3.2.1 Non-functional requirements

User friendliness: The system should be user-friendly, easy to understand, and easy to use.

Privacy: The system should be able to protect the privacy of users.

Availability: The system should be accessible to all users 24 hours a day.

Performance: The system should have fast performance and be able to respond to requests within a reasonable amount of time.

Accessibility: Users can access results from anywhere as long as they are within network service coverage. **Recoverability:** The system should be able to recover from all errors.

Environment: The system must run the Android operating system.

Security: Provide appropriate security to protect user information.

3.3.3 Functional Diagram

The functional diagram, also known as the system architecture diagram or system flow diagram, provides an overview of the major components and how they interact in an online wedding venue reservation management system. Please keep in mind that the following design is only a simplified example and the actual architecture may vary depending on the requirements and technologies used.

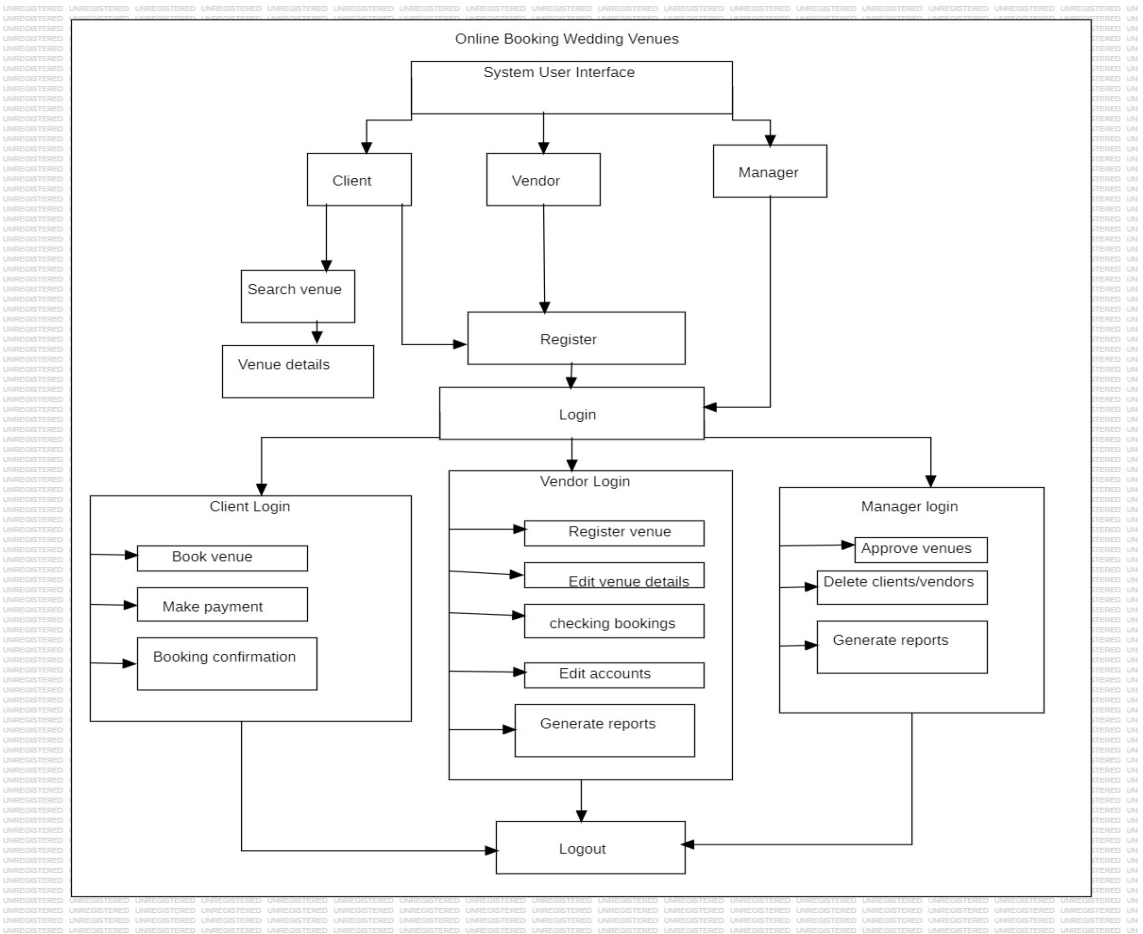


Figure 3.1 Functional Diagram

3.3.4 Methodological approach

The following below are some of the technologies and tools that have been used in the development of the system:

Visual Studio Code: a redesigned and enhanced code editor for creating and debugging contemporary online and cloud apps.

MongoDB: is a NoSQL database system widely used for storing and managing large volumes of document-oriented data.

Node.js: is an open-source, cross-platform JavaScript runtime environment that allows you to run JavaScript code outside of a web browser.

Express.js: is a popular **backend web application framework** built on top of Node.js. It provides a robust set of features and functionalities to simplify the process of developing web applications and APIs using JavaScript.

CSS (Cascading Style Sheets): is a style sheet language used for describing the presentation of a document written in a markup language.

HTML (Hypertext Markup Language): is the standard markup language for creating web pages.

3.3.4.1 Data collection techniques

Research methodology refers to the specific procedures or techniques used to identify, select, process, and analyze information about a topic to ensure that the results are valid and reliable, addressing the research objectives (Creswell, 2014). Depending on the researcher's plan and design, there are several ways data can be collected. The most commonly used methods include published literature sources, surveys (email and mail), interviews (telephone, face-to-face, or focus group), observations, and documents and records (Sekaran & Bougie, 2016).

a) Documentation

Documentation, as used here, involves collecting data from secondary sources such as books, reviews, class notes, and web pages to gather relevant information about online booking systems. This method ensures that the researcher has a comprehensive understanding of the existing literature and current practices in the field Corbitt, B. (2004)

b) Interviews

An interview, as the name implies, is a way of collecting data through direct conversation. In a research interview, both the interviewer and the interviewee participate in the conversation and answer the interviewer's questions. This helps in exploring study topics and improving comprehension. After speaking with a few potential users and venue owners, I was able to gather significant insights into the needs and challenges faced in the current booking process, which helped shape the development of the new system (Ross, 2003).

3.3.4.2 Software Development Methodology

The entire deployment process, which consists of a number of connected actions with potential transitions between them, is what makes a software system accessible to users. These actions may take place on either the producer or consumer side, or even on both. The specific processes or methods inside each activity can scarcely be stated because every software system is different. In light of this, "deployment" should be understood as a broad procedure that needs to be tailored to meet particular requirements or characteristics. (Michael C,2018)

a) Agile Software Development

Agile software development is a set of principles and practices for software development that emphasize flexibility, collaboration, and customer satisfaction. Agile methods are iterative and incremental, focusing on delivering small, workable pieces of software frequently, rather than a complete product at the end. This approach enables teams to respond to change quickly and effectively (Beck et al., 2001).

Common Agile Methodologies:

Kanban

Kanban methodology is about day-to-day workflows and processes. It is a simple, visual means of managing projects that enables teams to see the progress so far and what’s coming up next. Kanban projects are primarily managed through a Kanban board, which segments tasks into three columns: “To Do,” “Doing,” and “Done” (Anderson, 2010).

Scrum

Scrum is one of the most popular Agile methodologies, as it can bring teams together with a sharp focus and an efficient, collaborative approach to task execution. Sprints are the Scrum way of breaking projects down into iterations that can last anywhere between one and four weeks each. Bringing team members together from different departments, these sprints help you channel a collective focus to your projects (Schwaber & Sutherland, 2017).

3.3.4.3 System Design Methodology

System design is the first stage of the system development life cycle during which the user and you create a detailed knowledge of how the system will function. To fulfill the required requirement, the system's architecture, modules, interfaces, and data must be defined. (Steven D., 2016).

As a result of designing an application, various categories of information systems development methodologies are taken into consideration. Each information systems development methodology, however, has a unique manner for tackling each stage of the ISDM process. Agile, SSM, SSADM, RAD, PRINCE, Spiral, SDLC, Waterfall, Object-oriented, and XP are some examples of information system development methodologies that are available. (Banks & Wood-Harper, 2005). In our system development we will be dealing with Object Oriented Methodology.

A system development methodology called object-oriented methodology (OOM) promotes and facilitates the reuse of software components. This methodology permits the effective reuse of existing components and the sharing of its components by other systems, allowing a computer system to be constructed on a component basis.

Classes and objects are the two main aspects of object-oriented programming. A class

creates a new type (of object) where objects are particular instances of a particular class.

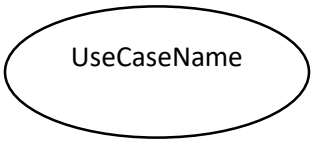
Use case

Use Case is a technique used in system analysis to locate, describe, and arrange the needs of the system. It serves as a representation of the system's functionality from a distance. A use case specifies a system function that gives an actor an observable consequence. The entity that interacts with the system is known as an actor (user, another system). It can be viewed as a collection of potential outcomes connected to a specific objective (Cockburn, 2001).

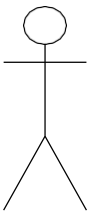
The use case diagrams are employed to compile the system's requirements. Actors, use cases, and their relationships make them up. The application's system or subsystem is modeled using the diagram. A single use case graphic illustrates a specific system functionality.

Elements of a Use Case Diagram

Use Case: Use cases serve as a proxy for what actors (users or other systems) want the system to accomplish. They define specific system functionalities and are depicted as ovals labeled with the use case name. This representation helps in capturing and organizing the functional requirements of the system (Pender, 2017).

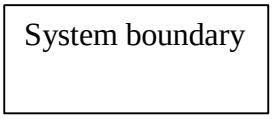


Actor: An actor specifies a role played by a user or any other system that interacts with the subject.



actorName

System Boundary: The system boundary is shown by a rectangle box drawn around all of the use cases to illustrate the system's borders or boundary. It details the system's scope and separates it from external entities (Booch et al., 2017).



Relationship between Use Cases

Association: a line between an actor and a use case

Include: The "include" relationship shows that the behavior of the included use case is part of the including (base) use case. The base use case is incomplete without the included use case, indicating that the included use case is mandatory for the base use case to function properly (Harrison, 2016).

<<include>> ----------->

Extend: The "extend" relationship specifies that one use case (the extending use case) extends the behavior of another use case (the base use case). The extending use case is typically optional, and the base use case must be meaningful and functional on its own (Arlow & Neustadt, 2017).

<<extend>> ----------->

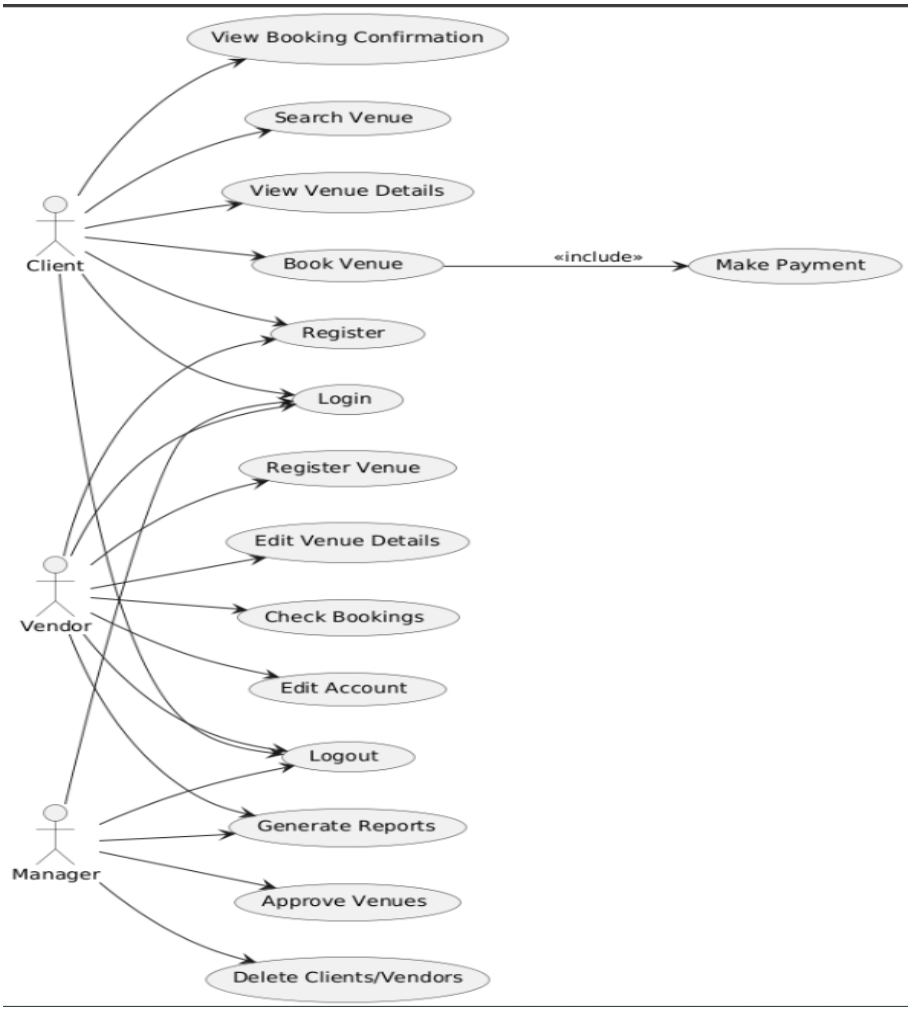


Figure 3.2. Use Case Diagram

Class diagram

A class diagram is a static model that displays the classes and connections between classes that hold true throughout the life of the system. The class diagram shows the relationships between classes, which comprise both behaviors and states. A class diagram's main objective is to display the connections between classes, also known as associations. The class, which stores and controls data in the system, is the primary component of a class diagram. (Wiley, 2015)

Components of a class diagram

Class Name
.- Attributes. Name
+ Operation. Name ()

Class: A class is a model for defining objects and behaviors within a system. In UML, a class describes a single item or a group of objects that share similar behaviors and structures. It is represented by a rectangle with rows for the class name, its attributes (characteristics), and its methods (operations) (Larman, 2017).

Attributes: The attributes of a classifier's instances are the details, facts, or characteristics that belong to them. A classifier can have any number of attributes or none at all. Instances of the classifier can hold attributes that describe a value or a range of values. An attribute's type, such as an integer or a Boolean, as well as its initial value, can be specified. Additionally, constraints can be added to attributes to define the range of possible values for them (Lang, 2019).

Operation: are the processes that a class knows to carry out. Operations most obviously correspond to the methods on a class.

Relationship between classes

In UML, an association between model components is referred to as a relationship. A UML relationship specifies the structure and behavior between model elements, thereby providing meaning to a model. These relationships are crucial for defining how different components of a system interact and collaborate (IBM Corporation, 2018).

Generalization

A relationship that bases one model element (the child) on another model element is known as a generalization relationship (the parent). In class, component, deployment, and use-case diagrams, generalization relationships are used to show that the child obtains all of the traits, operations, and relationships defined in the parent.

Association

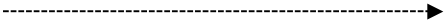
Association relationship represents a general binary relationship that describes an activity between two classes

Aggregation

This relationship is a unique kind of association that depicts the ownership connection between two groups. The link is modeled using aggregates such as has-a, part-of, owns, and employed- by.

Composition: Composition relationship is a special type of aggregation where parts are destroyed when the whole is destroyed.

Dependency: Dependency relationship exists between two classes if changes to the definition of one may cause changes to the other.



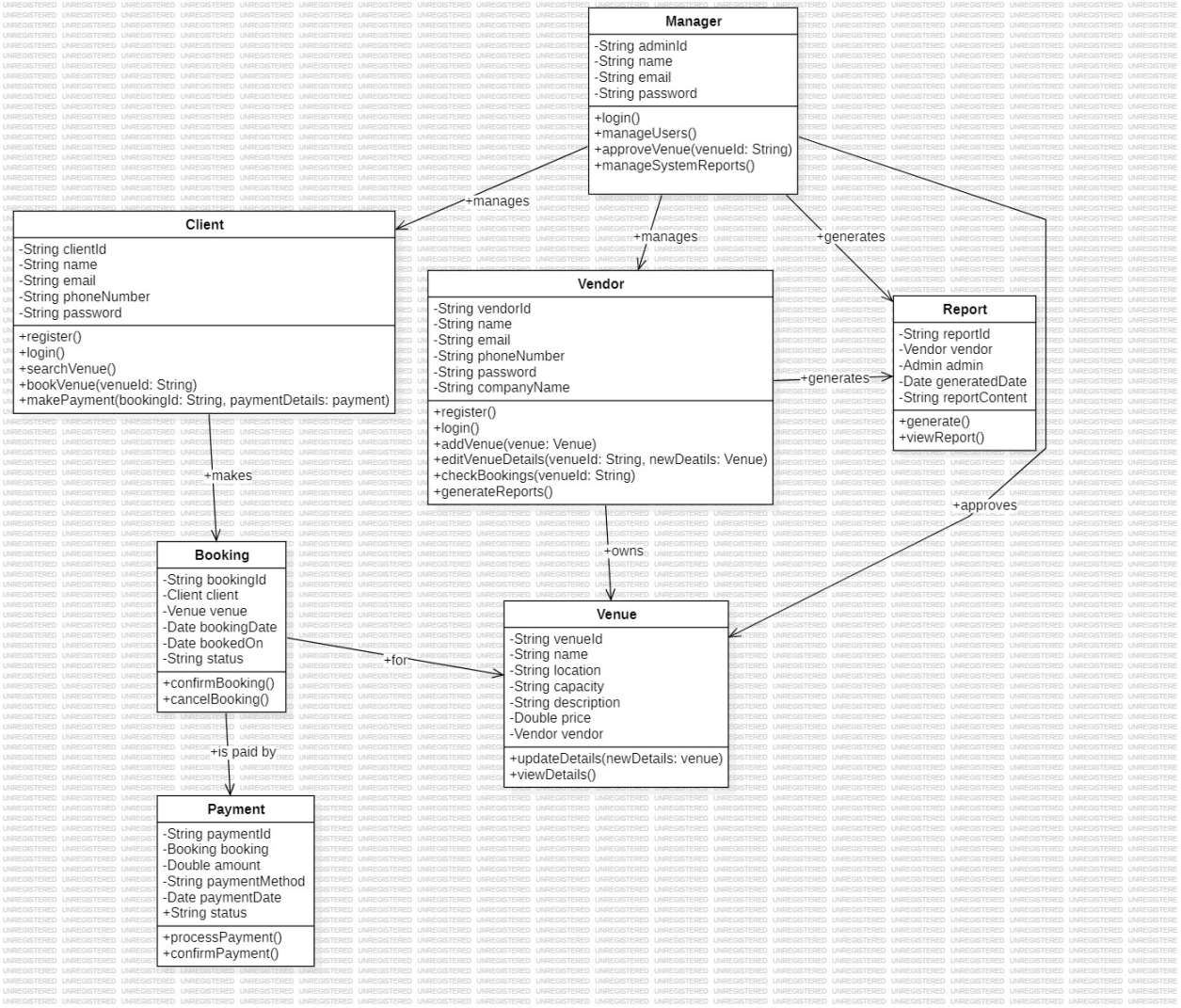


Figure 3.3 Class Diagram

Sequence diagram

A sequence diagram is a type of interaction diagram that illustrates the relationships between objects in a group and the sequence in which they interact. These diagrams are used by software engineers and business experts to understand the specifications of a new system or to describe existing processes. Sequence diagrams are also known as event diagrams or event scenarios (Rumbaugh et al., 2017)

The following scenarios are ideal for using a sequence diagram:

Usage scenario (we will be using on our system): A usage scenario is a schematic showing possible applications for our system. It's an excellent technique to ensure that you have considered the logic of each possible usage situation for the system.

Method logic: Just as you might use a UML sequence diagram to explore the logic of a use case, you can use it to explore the logic of any function, procedure, or complex process.

Service logic: If you consider a service to be a high-level method used by different clients, a sequence diagram is an ideal way to map that out.

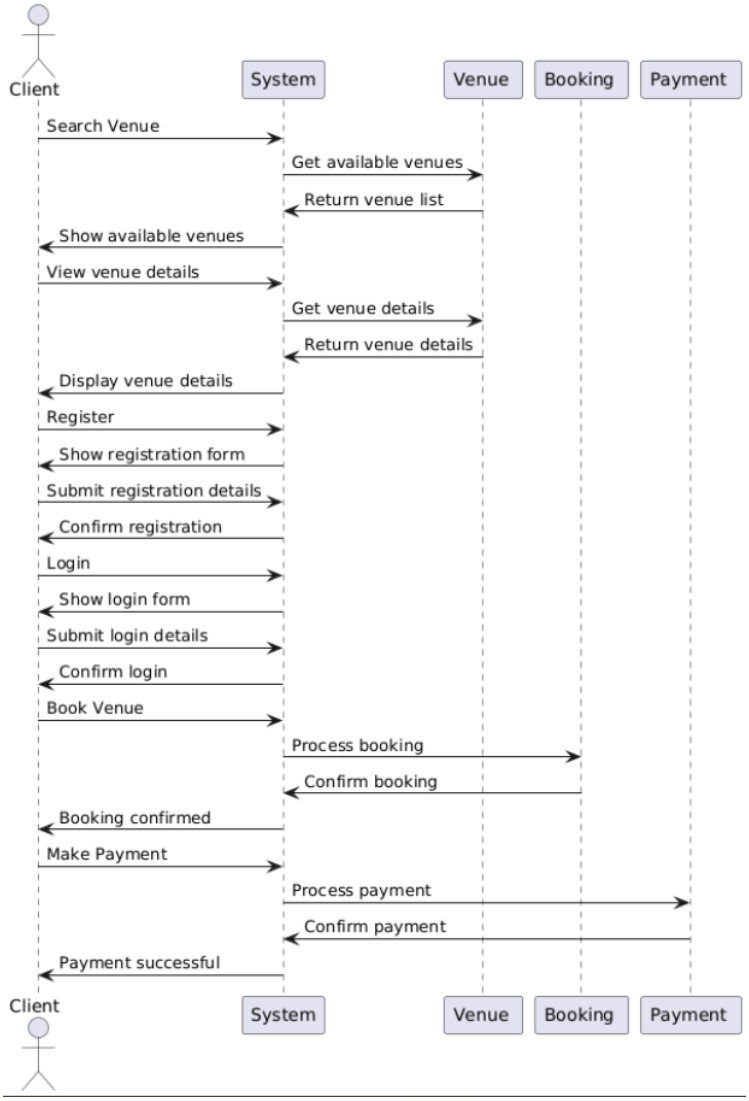


Figure 3.4 Client Sequence Diagram

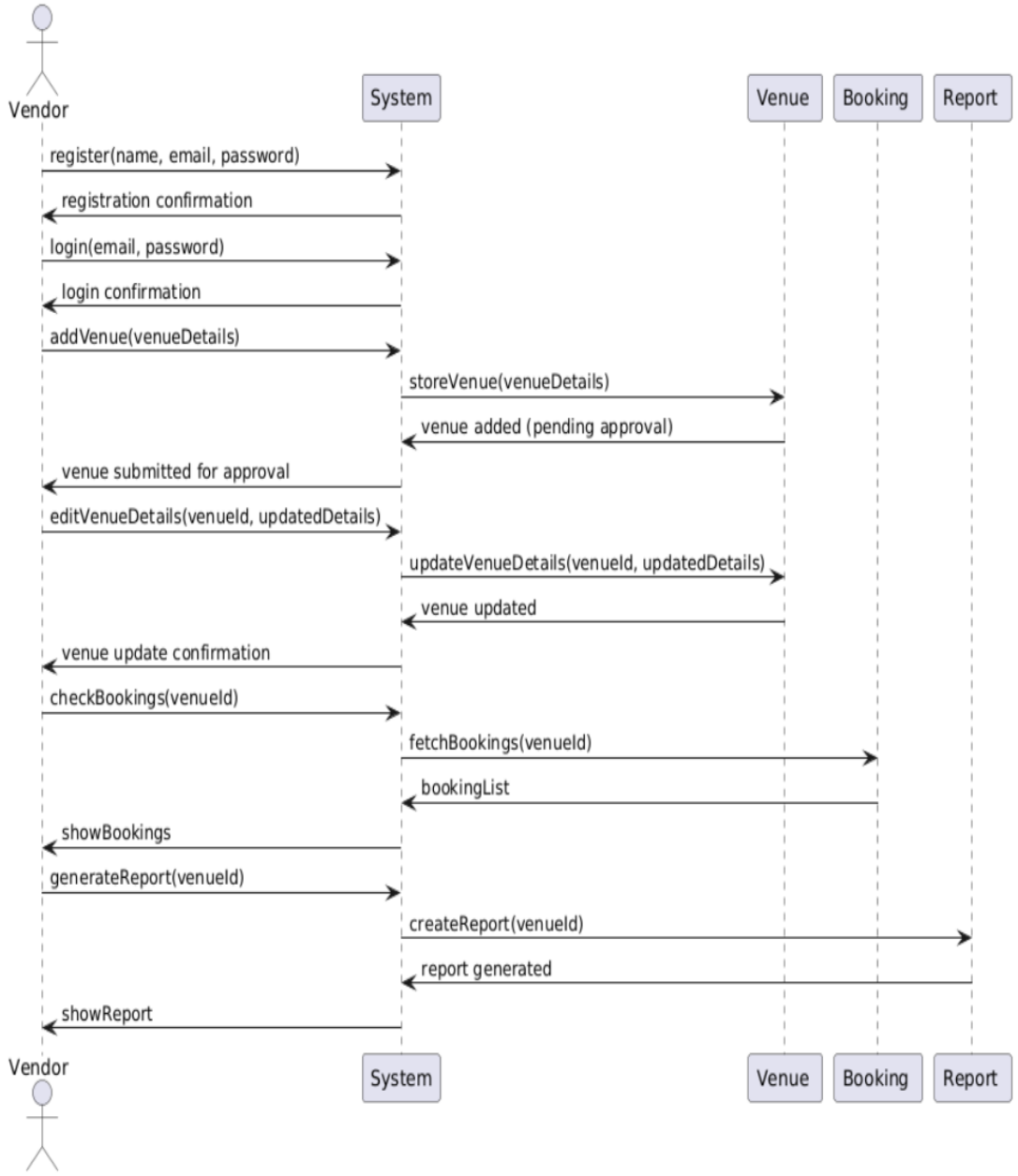


Figure 3.5 Vendor Sequence Diagram

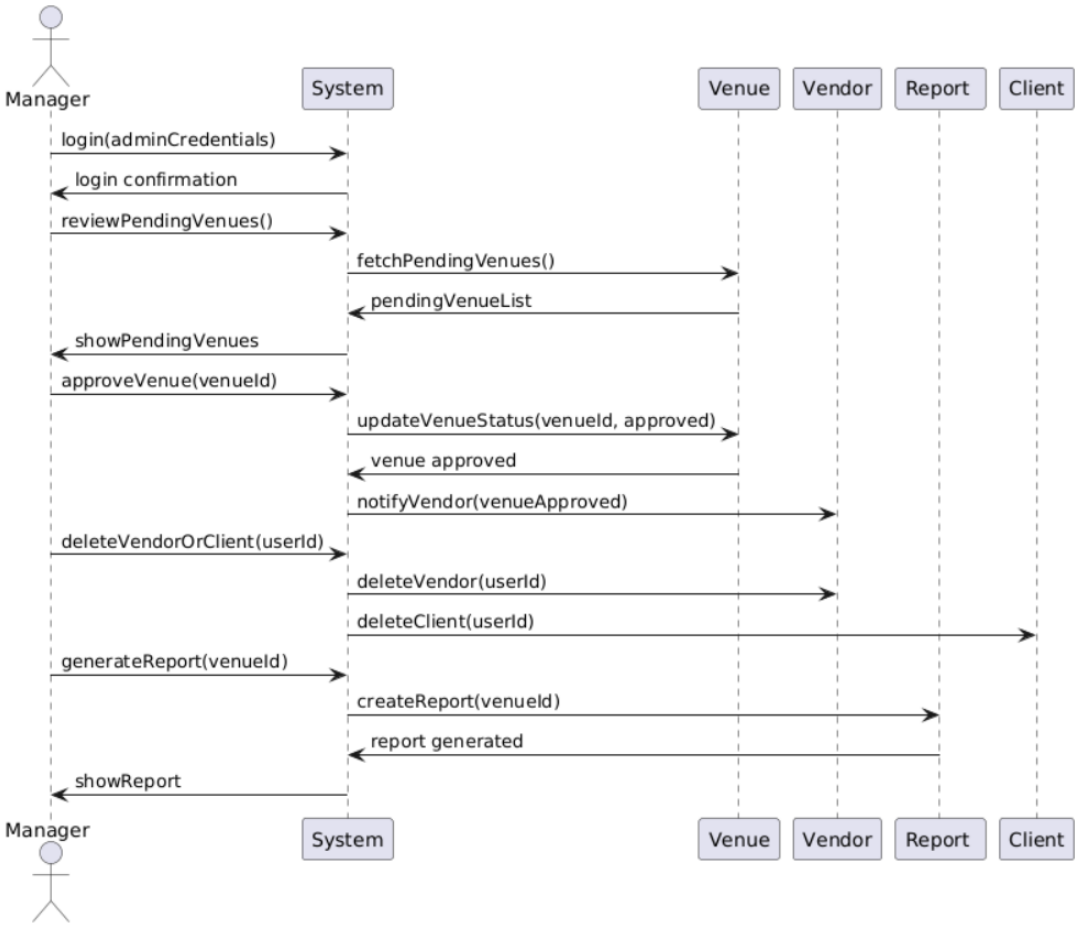


Figure 3.6 Manager Sequence Diagram

CHAPTER IV: SYSTEM IMPLEMENTATION

4.1. Implementation and Coding

4.1.1. Introduction

The Online Booking Wedding Venue Management System implementation stage refer to a process of translating the design specifications into a working system. This chapter gives details of coding practices, tools, and technologies that were used during the stage of implementation.

4.1.2. Tools and Technologies Used

1 Backend Development

- **Node.js**: The runtime environment for executing JavaScript on the server side.
- **Express.js**: A web application framework for Node.js, used for handling routing, middleware, and server logic.
- **Passport.js or JWT**: For user authentication and authorization. Can be used to manage sessions for Client, Vendor, and Admin logins.
- **Multer**: Middleware for handling file uploads (e.g., when vendors upload images of venues).
- **Bcrypt.js**: For securely managing user credentials during registration and login.
- **NodeMailer**: For sending email confirmations for bookings, payments, or vendor registration approvals.
- **dotenv**: For managing environment variables like API keys, database URIs, and secret keys in a secure way.

2. Frontend Development

- **HTML5**: Used for structuring the web pages and forms for registering, booking, and logging in.
- **CSS3**: For styling the web pages, enhancing the look and feel of the user interface (UI).
- **JavaScript :** For client-side interactivity, validation, and UI dynamics.
- **EJS** (Embedded JavaScript): A templating engine used with Express to render HTML with dynamic content (like venue listings, bookings).

3.Database

- **MongoDB:** MongoDB is a NoSQL database that stores data in flexible, JSON-like documents, making it particularly well-suited for applications with unstructured or semi-structured data, such as user profiles, venue details, and booking records (Chodorow & Dirolf, 2019).
- **Mongoose:** Mongoose is an Object Data Modeling (ODM) library for MongoDB and Node.js. It provides a schema-based solution to model application data and manage relationships between entities, simplifying data interactions and validations (Hodges, 2021).4. Version Control

Git: Git is the most widely used version control system. It enables you to track changes, roll back to previous versions, and manage branching for features, bug fixes, and releases, facilitating collaborative development and efficient version management (Chacon & Straub, 2014).

GitHub: Platforms for Hosting my git Repository.

5. **Development Tools**

- **VS Code**: A popular code editor with extensions that help in Node.js and MongoDB development.

6. Deployment

Render: The application is hosted on Render, a cloud-based deployment platform that supports Node.js applications**.** Render provides auto-deployment from connected GitHub repositories, SSL certificates, scalability, and simplified management of environment variables.

4.1.3. Screenshots

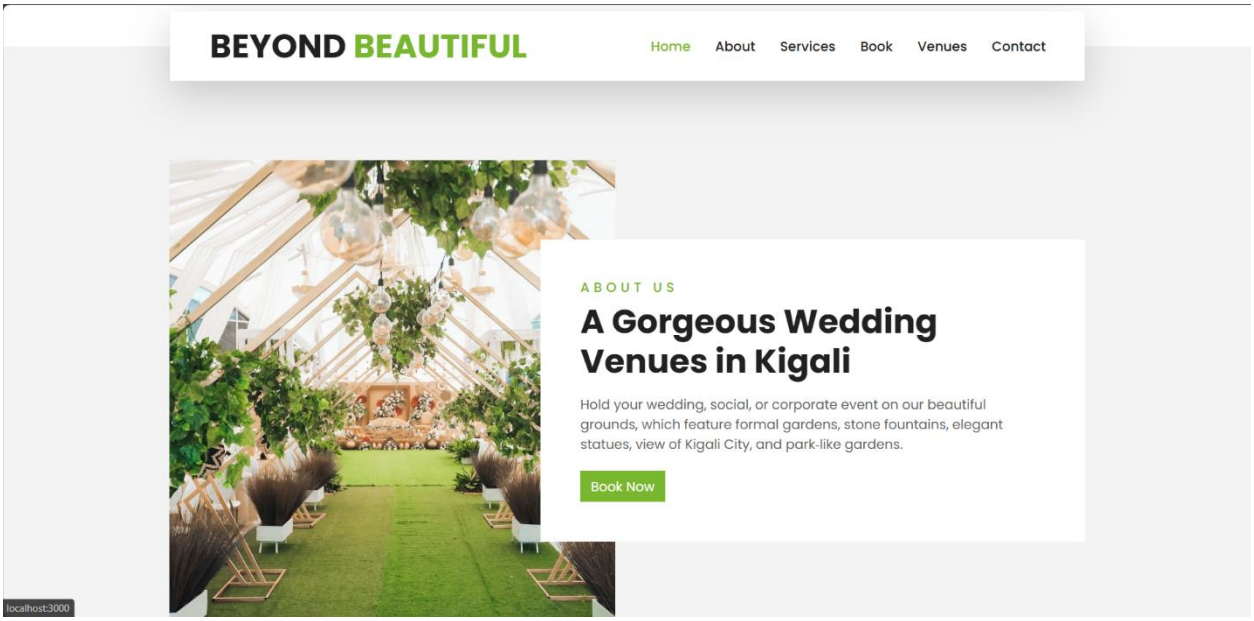


Figure 4.1 Home Page

This is the first page users see when they visit the **Online Wedding Venue Booking System**.

Where client can be able to explore venues registered and see their details.

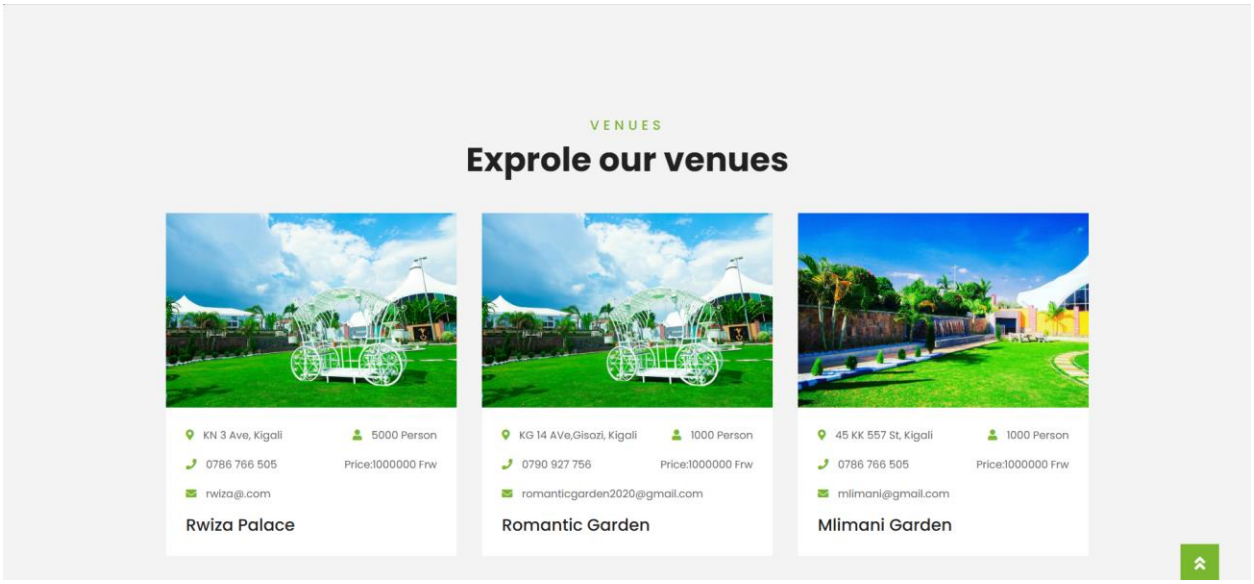


Figure 4.2 Venues page

This is the venues page; clients can view a list of all available venues. If they want more details about a specific venue, they can click on it to see more information.

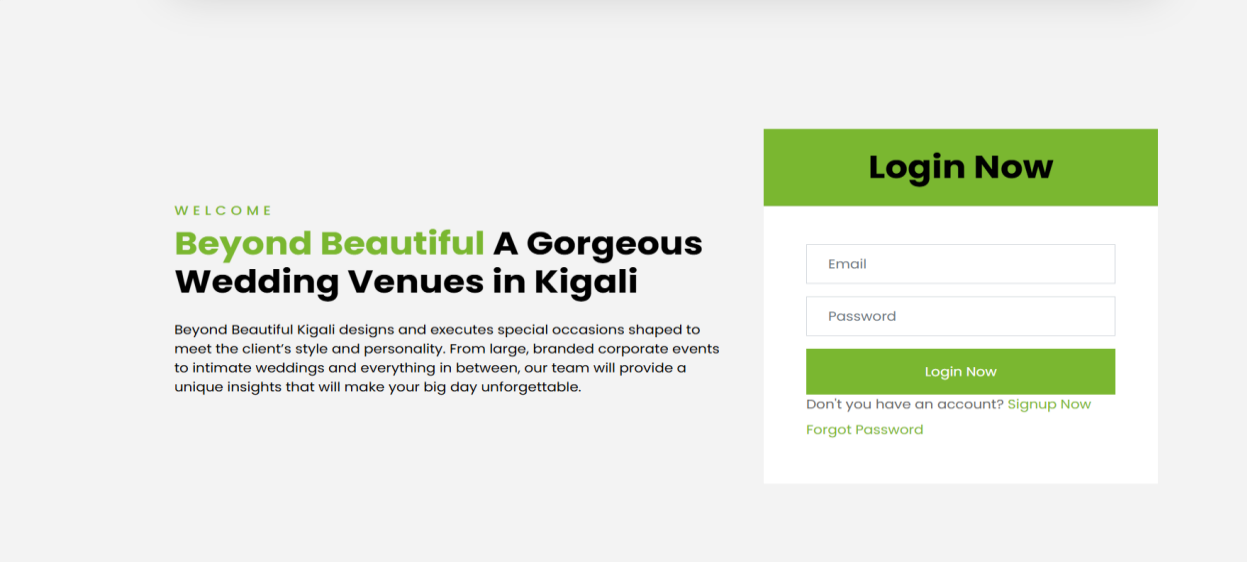


Figure 4.3 Login Page

This is a login page where a user will enter his email and password to get access to his account and manage it. If he doesn't has an account he can click on sign up to create an account and if he has forgotten his password he can click on forgot password where it will redirect to forget password page and enter his email to receive a link for resetting the password.

4.1.3.1. Client side

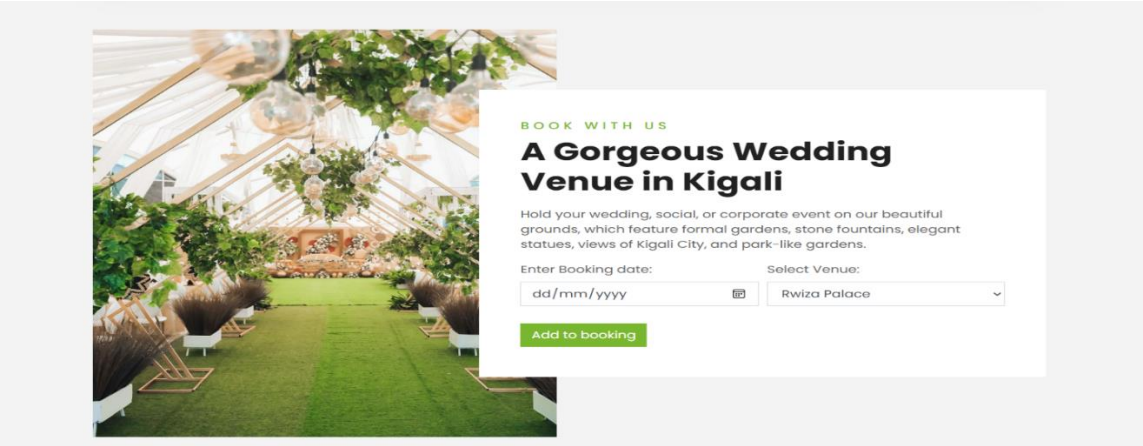


Figure 4.4 Booking Page

This is the page Client will see after login, is the booking page where client can be able to select dates and venue they want if the date is already booked they will get a message telling them the date is not available. After successful booking they can go to the page containing their booking.

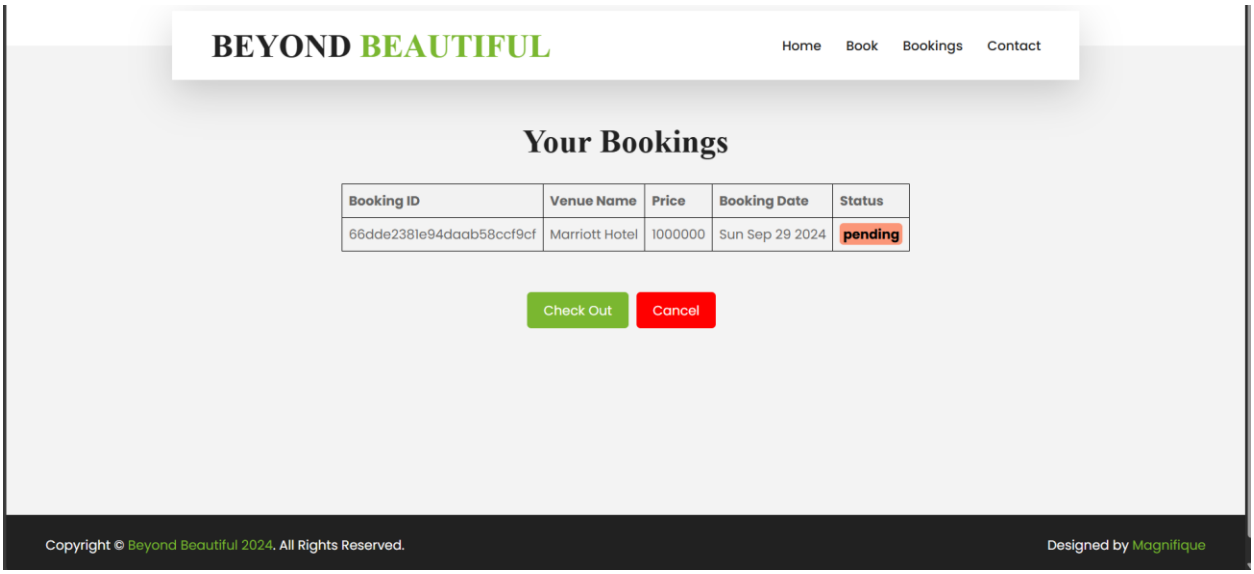


Figure 4.5 Booking Results

After booking this is the page your see, it contains the information of your booking, the status is pending it will be approved after paying.

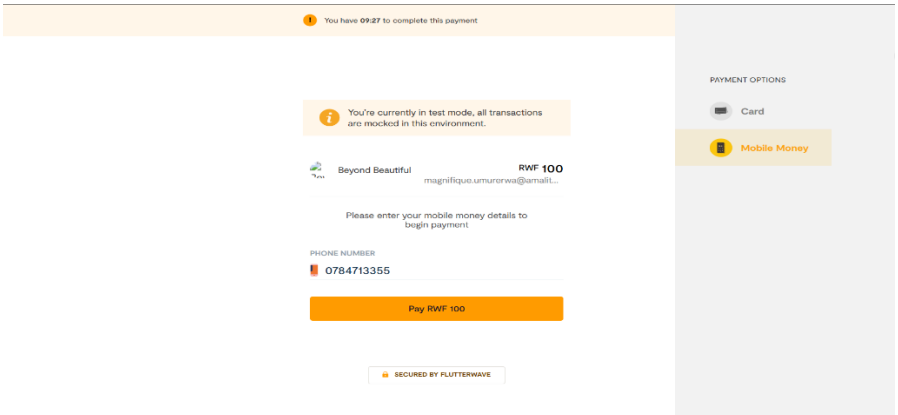


Figure 4.6 Payment Page

For payment I used Flutterwave APIs, that form also is provided by them. By integrating my system with their payment APIs client can be able to pay but It's in testing mode it's not a really payment money cannot be deducted from your account.

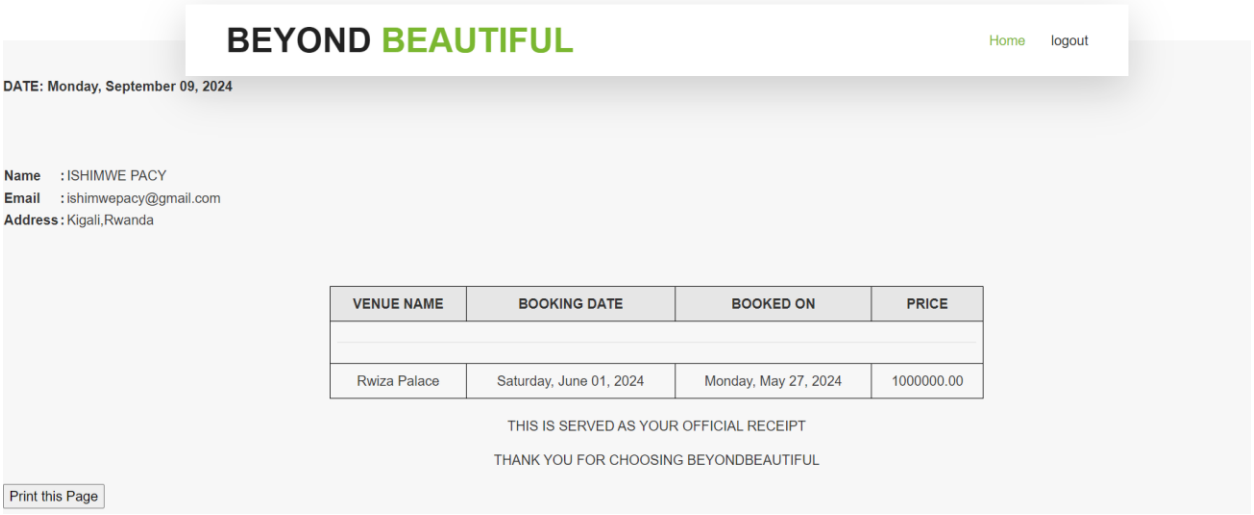


Figure 4.7 Receipt

After successful payment this is the page the client will see as a receipt and he can print it if he wants.

4.1.3.2. Vendor side

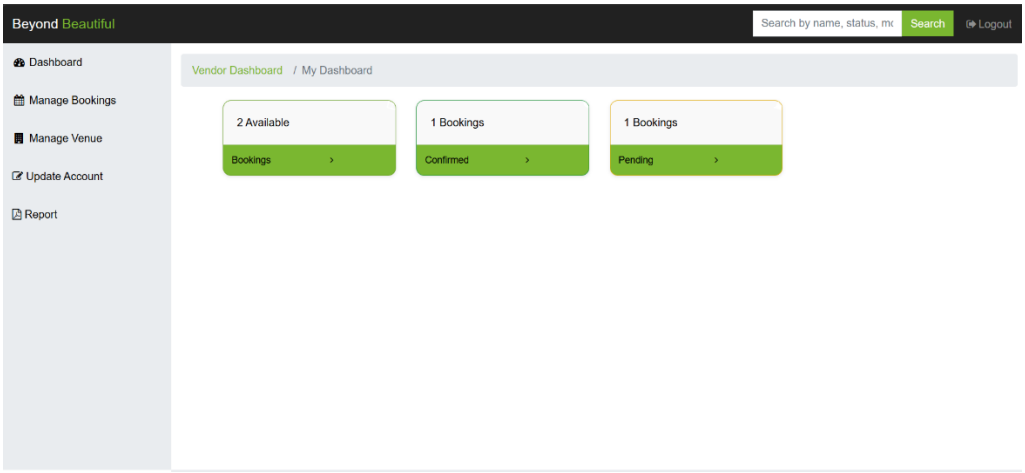


Figure 4.8 Vendor Dashboard

This is the dashboard of the vendors where they can be able to manage their venues. The page shows bookings for their venues, including available, confirmed, and pending ones.

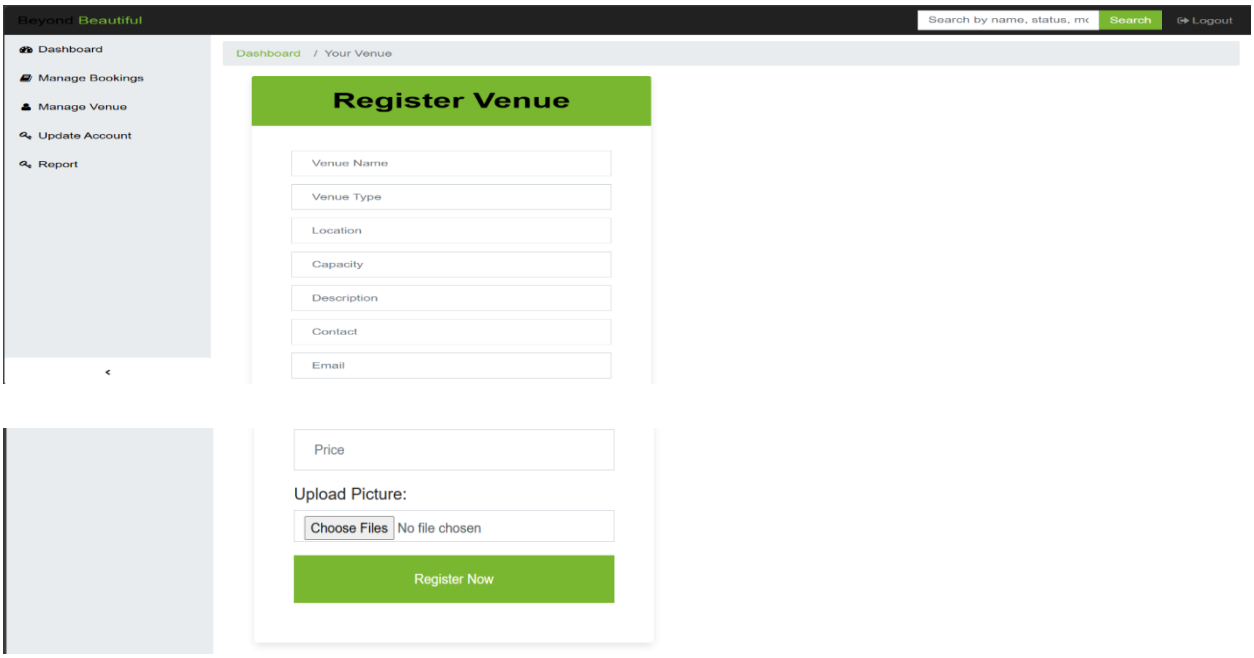


Figure 4.9 Register Venue

If you are new to the system as vendor, after registering, you can add your venue and then wait for the manager to approve it.

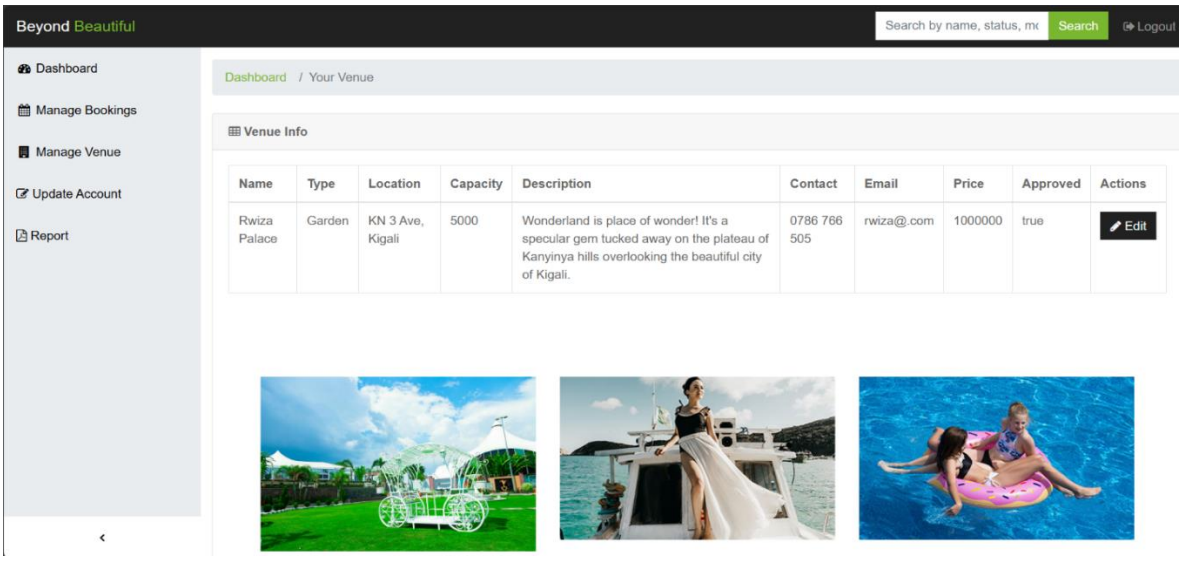


Figure 4.10 Vendor Venue Page

This page contains the information of venue but each Vendor can see information of his venue only the one he registered and he can be able to manage it like editing.

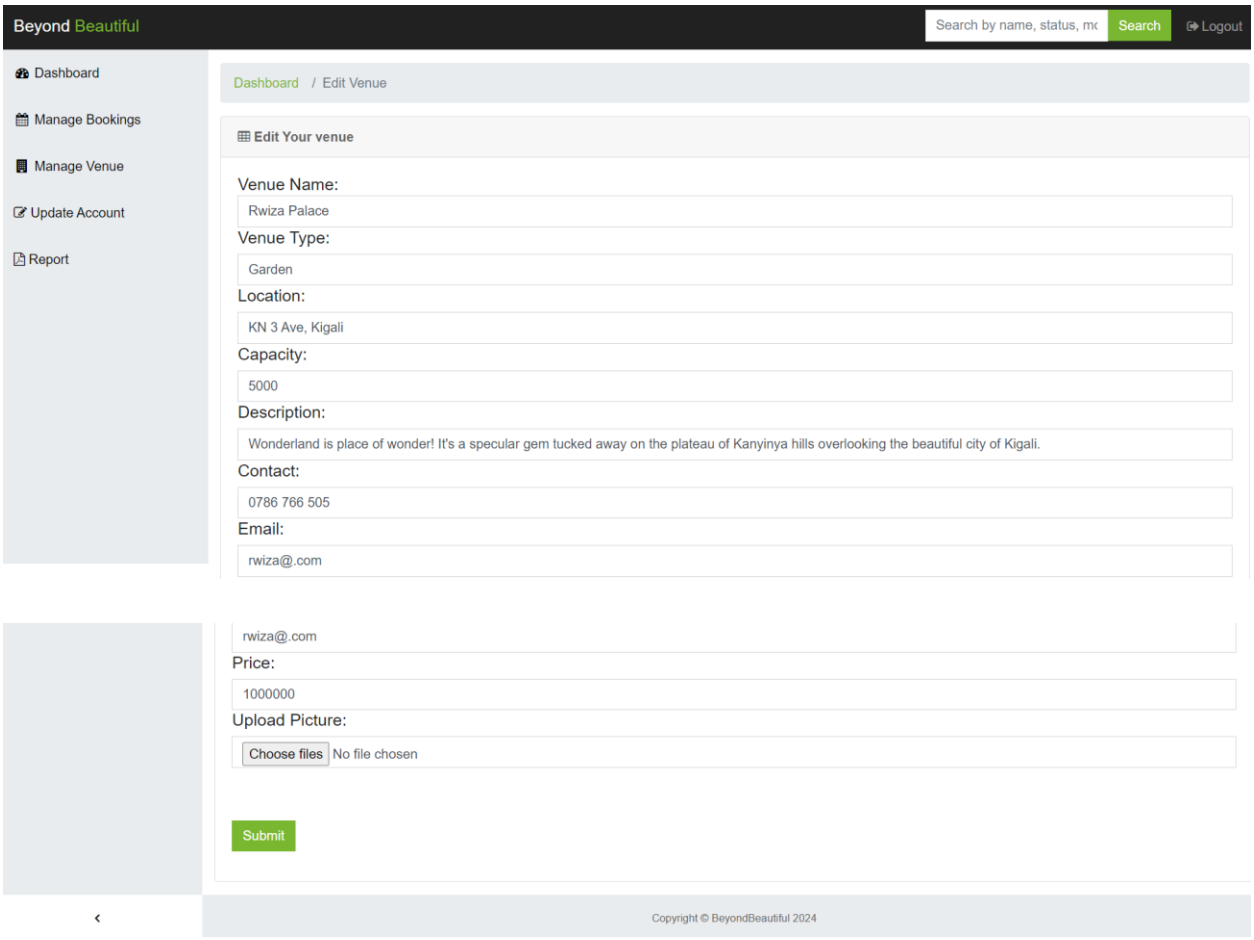


Figure 4.11 Venue Edit Page

On the venue edit page, you can update your venue's information or upload new images.

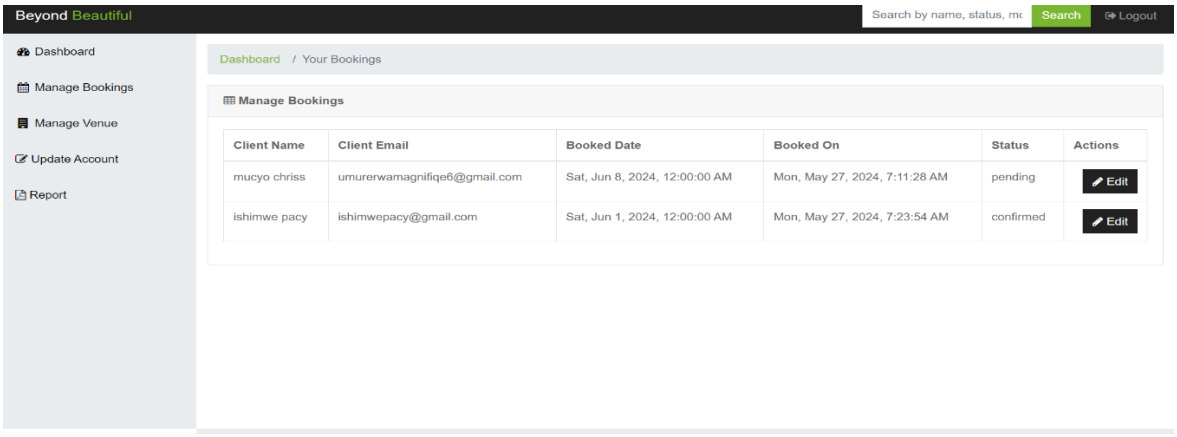


Figure 4.12 Venue Bookings

This page shows the bookings for your venue. You can check their status to see if they're confirmed or not. You can also edit bookings if a client wants to change the dates.

4.1.3.3. Manager side

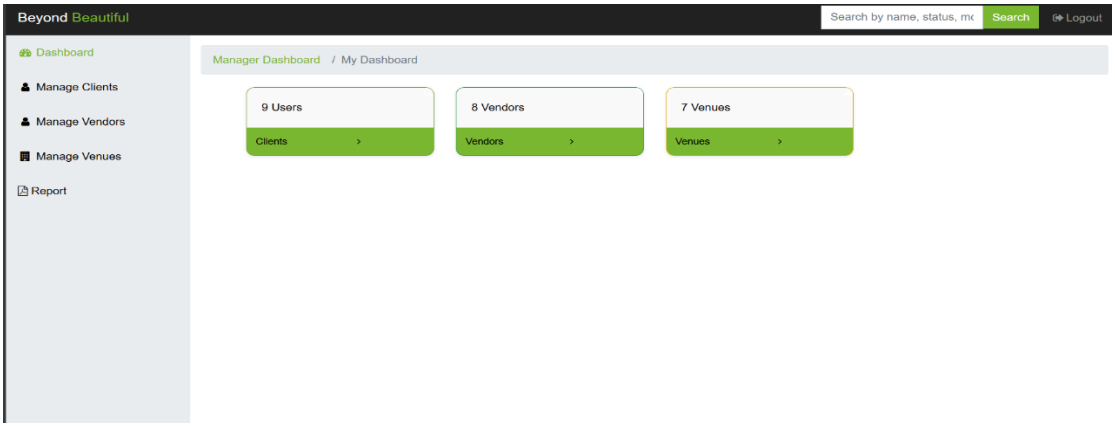


Figure 4.13 Admin Dashboard

This page is the dashboard of the manager where he can be able to control and manage the system. On dashboard he can see the client’s system have, the vendors and venues.

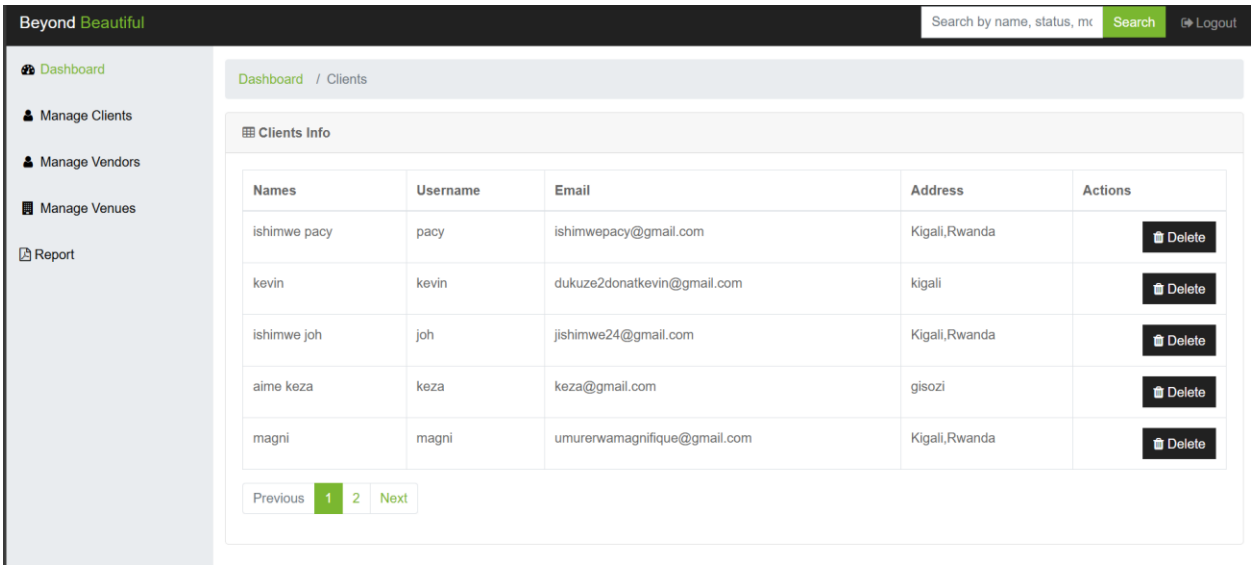


Figure 4.14 Manage Clients

On this page, the manager can view all clients and manage them, including deleting those who misuse the system.

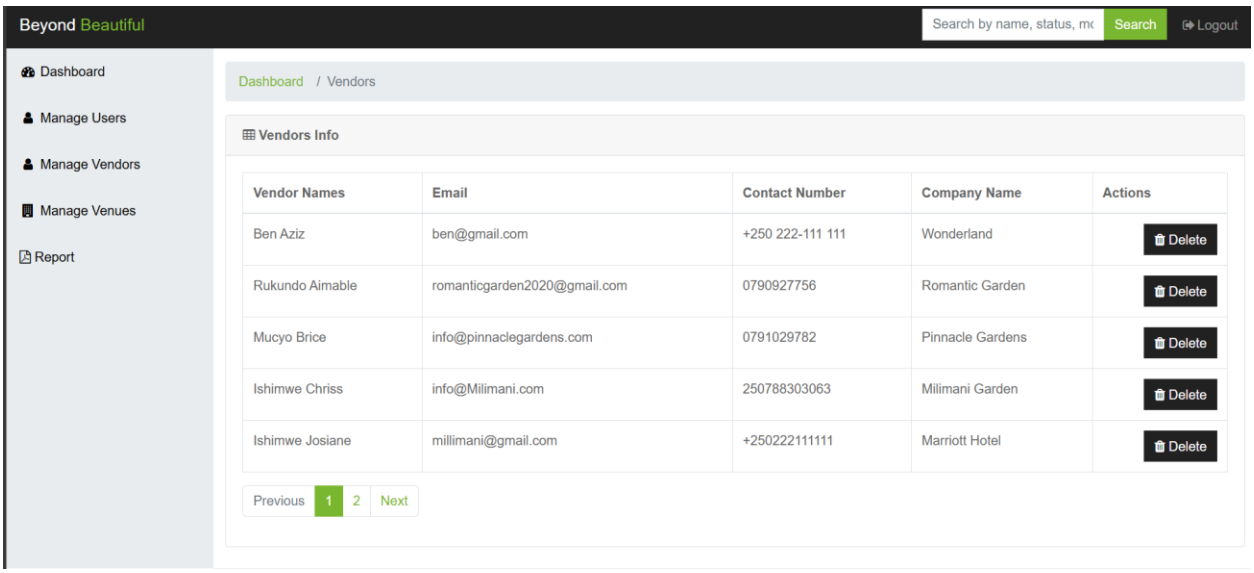


Figure 4.15 Manage Vendors

On this page, manager can view all vendors and manage them, including deleting those who doesn't follow the rules of the system and deleting vendor it deletes his venue also.

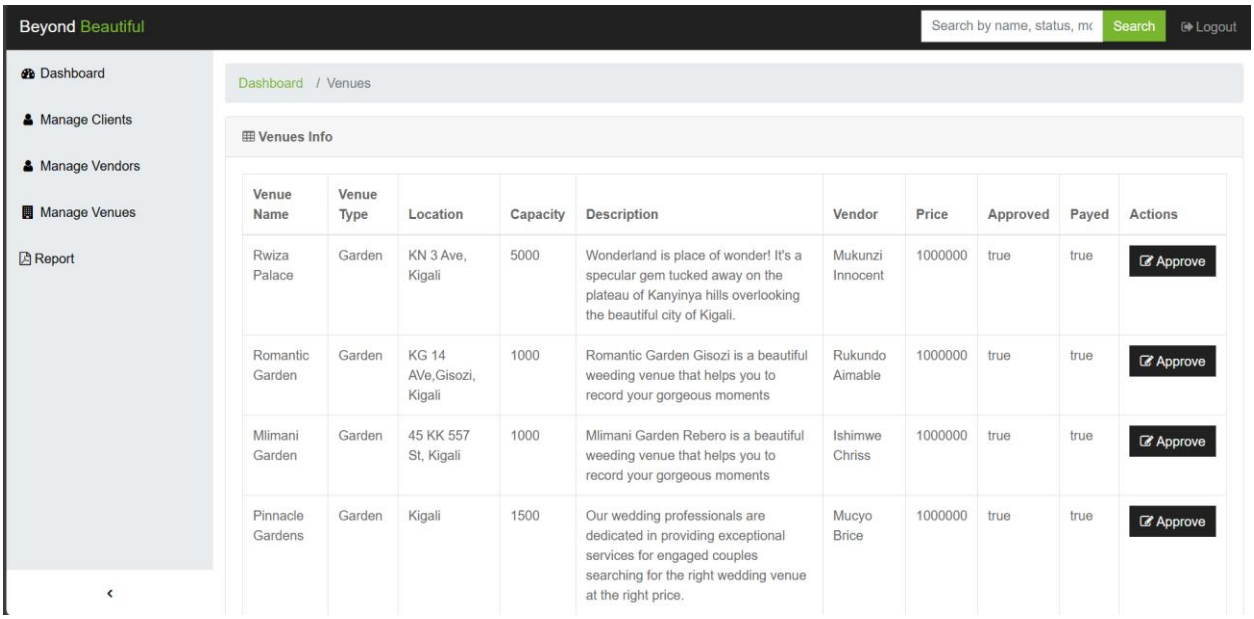


Figure 4.16 Manage Venues

On this page, manager can view all venues and manage them by approving them after their registration and he approves them after vendor of that venue the one who registered it made payment.

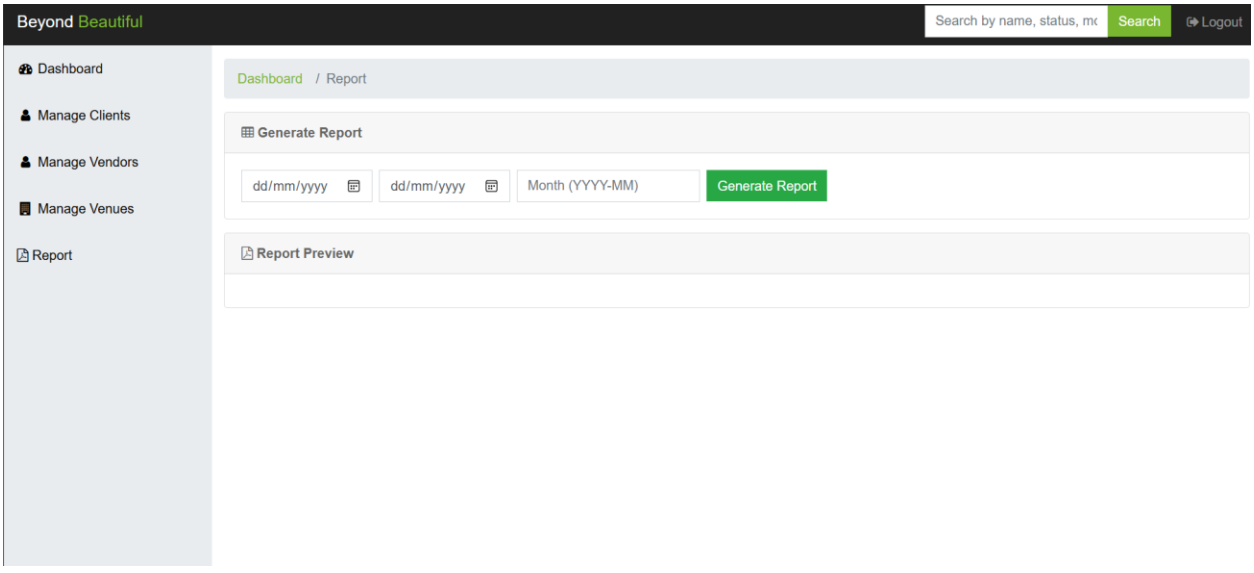


Figure 4.17 Report

This page is used to generate reports, you can generate reports whether using range of dates or using month name.

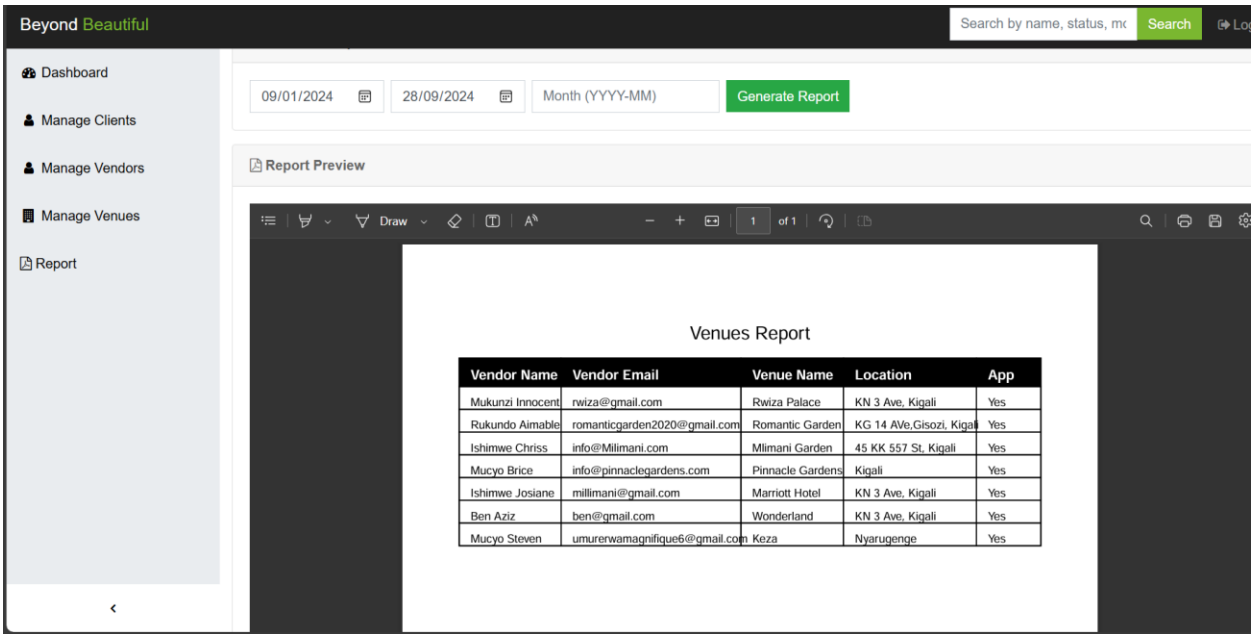


Figure 4.18 Report Result

After generating report, that's the report preview and you can save it as PDF file to your PC if you want.

4.1.4. System Source Code

```
async function createUser(req, res, next) {
    const data = req.body;
    const existingUser = await User.findOne({ email: data.email });
    if (existingUser) {
        return res.redirect('/signup?error=Email already exists');
    }
    const otp = otpGenerator();
    const otpExpirationDate = new Date().getTime() + (60 * 1000 * 5);
    const user = await User.create({
        names: data.names,
        username: data.username,
        email: data.email,
        password: data.password,
        address: data.address,
        otp: otp,
        otpExpires: otpExpirationDate,
    });
    const mailOptions = {
        from: '"Beyond Beautiful" <ketymagnifique@gmail.com>',
        to: user.email,
        subject: 'OTP Verification',
        text: `Dear ${user.username},\n\nWelcome to our website! Your account has been successfully created.Your OTP: ${otp}`
    };
    transporter.sendMail(mailOptions, (error, info) => {
        if (error) {
            console.error('Error sending email:', error);
        } else {
            console.log('Email sent:', info.response);
        }
    });
    return res.redirect('/otp?error=Check your email to verify your account');
  } catch (error) {
    next(error);
  }
}
```

Figure 1 Register backend Codes

Those are the codes to create account, firstly it checks if the account exits, if exists it gives you message saying “email already exists” if not you proceed with creating account by providing your information after clicking submit it sends email to use for verifying account.

```
async function createBooking(req, res, next) {
    try {
        const userId = req.session.userId;
        const { bkng_date, venue_name } = req.body;
        if (!userId || !bkng_date || !venue_name) {
            return res.redirect('/book?error=User ID, booking date, and venue name are required.');
        }
        const bookingDate = new Date(bkng_date);
        const currentDate = new Date();
        currentDate.setHours(0, 0, 0, 0);
        if (bookingDate <= currentDate) {
            return res.redirect('/book?error=Booking date must be a future date.');
        }
        const hasActiveBooking = await Booking.exists({ user: userId, status: { $in: ['pending', 'confirmed'] } });
        if (hasActiveBooking) {
            return res.redirect('/bookings?error=You already have an active booking. Please complete or cancel your current booking before making anc
        }
        const foundVenue = await venue.findById(venue_name);
        if (!foundVenue) {
            return res.redirect('/book?error=Venue not found.');
        }
        const isDateBooked = foundVenue.bookedDates.some(date => {
            const bookedDateString = date.toISOString().split('T')[0];
            const bookingDateString = bookingDate.toISOString().split('T')[0];
            return bookedDateString === bookingDateString;
        });

        if (isDateBooked) {
            return res.redirect('/book?error=The venue is not available on the selected date.');
        }
```

```
        const booking = {
            user: userId,
            venue: foundVenue._id,
            bookingDate,
            status: 'pending',
            price: foundVenue.price
        };
        await Booking.create(booking);
        foundVenue.bookedDates.push(bookingDate);
        await foundVenue.save();

        res.redirect('/bookings');
    } catch (error) {
        console.error('Error creating booking:', error);
        next(error);
    }
}
```

Figure 2 Booking backend codes

This is the codes to make a booking with error validation like if you choose past date system will tell you “the date must be the future date”. And if the date is already taken you will get a message telling you that. After successful booking you will be redirected to the page contains your booking.

```
const start = async () => {
  try {
    await connectDB(process.env.MONGODB_URI);
    app.listen(process.env.PORT, () =>
      console.log(`Server is listening on port ${process.env.PORT}...`)
    );
  } catch (error) {
    console.log(error);
  }
};

start();
```

Figure 3 Server Codes

This the codes for starting my server and connecting to the database. Database url link and port number are defined in Dotenv file. Dotenv file is the file that contains environment variables I have used while building this system like passwords and all sensitive information. For example, when you are pushing your codes on GitHub and you have some passwords in codes it cannot allow you to publish them. That’s why it’s a good idea to define them in Dotenv file because this file is not pushed on GitHub.

4.2. Testing

4.2.1. Introduction

Testing is a crucial phase in the software development lifecycle, ensuring that the application behaves as expected and meets the specified requirements (Myers, Sandler & Badgett, 2011). The testing process involves validating individual components (unit testing), ensuring that these components work together (integration testing), and validating the system as a whole (system testing) (Pressman, 2014). I also performed validation, functional, and acceptance tests to confirm that the final product is ready for deployment (Black, 2009).

Some types of software testing:

4.2.2 Unit Testing

In this project, I performed unit testing on key components of the online wedding venue booking system. Specifically, I tested individual modules such as:

User Authentication: Ensured that the login and registration functions worked as expected by testing scenarios like valid/invalid credentials, password validation, and session management.

Booking Process: Tested the core booking functionality to ensure that selected venues could be successfully booked, and errors were returned when trying to book an already reserved venue.

4.2.3 Validation Testing

For Validation Testing, I validated the overall functionality of the system against the stakeholder requirements. This involved reviewing the entire booking flow, from user login and venue selection to payment processing, to ensure that the system behaved as expected in real-world scenarios. I conducted tests such as:

Checking if users could browse venues based on filters like date, location, and capacity.

Verifying if booking confirmations were correctly generated.

Ensuring that payment processing integrated smoothly with the third-party provider and returned appropriate responses for successful and failed payments.

4.2.4 Integration Testing

For Integration Testing, I tested the interactions between different modules to ensure they worked seamlessly together. Some of the key integrations tested were:

User Registration and Database Storage: Verified that new users could successfully register and their data was correctly saved in the MongoDB database.

Booking Process and Payment Gateway: Ensured that the booking system properly triggered payment requests, and confirmed that a successful payment resulted in booking confirmation.

4.2.5 Functional Testing System Testing

In Functional Testing, I tested the core functions of the system to verify whether it met the specified functional requirements. I created a series of test cases to evaluate both positive and negative scenarios:

Positive Tests: Successfully booking a venue after checking its availability, filtering venues by criteria (e.g., location, date), and processing payments.

Negative Tests: Attempting to book a venue with invalid credentials, trying to book a venue that was already reserved, and simulating failed payments.

4.2.6 Acceptance Testing

Acceptance Testing was performed after system testing to confirm that the system met the client's requirements and was ready for deployment. This included

Beta Testing: Released to a small group of users to gather feedback on the system’s usability, functionality, and performance in real-world conditions.

CONCLUSION AND RECOMMENDATIONS

4.7. CONCLUSION

The main objective of this project was to provide a solution to the challenges involved in booking wedding venues, such as the lack of transparency, inefficiencies in the booking process, and time management. With this system, users are able to search for, view, and book wedding venues online without needing to visit venues in person, streamlining the process and making it more convenient. Additionally, the system enables venue managers to manage their bookings, track venue availability, and handle customer requests in a more organized and efficient manner.

To achieve the goals of this project, various methods, technologies, and tools were employed. UML diagrams were used for the analysis and design of the system, ensuring a comprehensive structure. Observations and feedback from venue managers and customers were gathered to gain insights into the current challenges and to design features that meet their needs effectively.

The development of this Online Wedding Venue Booking System represents a significant improvement in the wedding planning process. It not only saves time for users but also provides venue managers with the tools they need to manage their businesses more effectively.

Finally, I remain open to receiving suggestions and feedback for future improvements to this system, as continuous refinement is essential to maintaining its usefulness and efficiency.

4.8. RECOMMENDATIONS

I would like to recommend that event management companies and local authorities, especially in Kigali city, consider adopting the Online Wedding Venue Booking System. Implementing this system in real life would significantly streamline the process of searching for, booking, and managing wedding venues. It would reduce manual paperwork, save time, lower administrative costs, and provide transparency throughout the booking process. By enabling users to make reservations online, the system would make the process more efficient and accessible to individuals planning events.

This system is scalable, with the potential to expand its features and functionalities as the needs of users evolve. It could also be adapted to manage bookings for other event types or facility rentals. In concluding this project, I would like to affirm that the Online Wedding Venue Booking System offers a practical and efficient solution to the challenges of venue booking. I encourage future developers and stakeholders to continue enhancing this system by adding new features to further improve its usability and service delivery.

4.9. Future Work

Looking ahead, there are several opportunities to improve the Online Wedding Venue Booking System. Developing a mobile app would make it easier for users to book and manage venues directly from their phones, while also allowing for timely notifications and updates. Adding artificial intelligence could personalize venue recommendations and automate some processes, making the system even more efficient. Enhancing the user experience with features like virtual venue tours and live chat support could help users make better decisions and get help when needed. Expanding the system to support multiple languages and currencies would make it accessible to a broader audience. Additionally, integrating with social media and event planning tools could streamline event planning and promote venues more effectively. These upgrades could significantly improve the system, making it more user-friendly and versatile.

References

- Davis, N. (2000). Bridal style. Southport, CT: Hugh Lauter Levin Associates.
- Daniels, M. J., Lee, S., & Cohen, T. (2012). "*The attributes influencing wedding reception venue selection. Event Management",* 16(3), 245-258. Retrieved from https://www.cognizantcommunication.com/journal-titles/event-management.
- Daniels, M., & Loveless, C. (2007). *"Wedding planning and management: Consultancy for diverse clients. Burlington",* MA: Buttersworth-Heinemann.
- Tonisha parra (2023). https://www.itbriefcase.net/venue-management-software-and-on-demand-reporting-making-informed-decisions.
- https://www.collinsdictionary.com/us/dictionary/english/wedding-venue.
- Alzua-Sorzabal A, Gerrikagoitia JK, Torres-Manzanera E (2013) "*Opening and closing Internet booking channel for hotels: a first approximation*". Tour Manag Perspect 5:5–9.
- https://www.trustpilot.com/review/weddingwire.com
- Stonebraker, M., & Hellerstein, J. M. (2005). "*What Goes Around Comes Around. In Readings in Database Systems"* (pp. 2-25). The MIT Press.
- Elmasri, R., & Navathe, S. B. (2017). Fundamentals of Database Systems (7th ed.). Pearson.
- Silberschatz, A., Korth, H. F., & Sudarshan, S. (2020). Database System Concepts (7th ed.). McGraw-Hill.
- Sadalage, P. J., & Fowler, M. (2013). NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence. Addison-Wesley.Codd, E. F. (1970). "*A relational model of data for large shared data banks. Communications of the ACM"*.
- Garcia-Molina, H., Ullman, J. D., & Widom, J. (2008). *"Database Systems: The Complete Book (2nd ed.)".* Prentice Hall.
- Norman, D. A. (2013). "*The Design of Everyday Things: Revised and Expanded Edition. Basic Books*".
- Newman, S. (2015). "*Building Microservices: Designing Fine-Grained Systems"*. O'Reilly Media.

- Booch, G. (1994). "*Object-Oriented Analysis and Design with Applications (2nd ed.").* Addison-Wesley.
- Evans, E. (2003). "*Domain-Driven Design: Tackling Complexity in the Heart of Software"*. Addison-Wesley.
- Corbitt, B. (2004). *E-commerce Systems Architecture and Applications*. Wiley.
- Booth, A., Sutton, A., & Papaioannou, D. (2016). Systematic Approaches to a Successful Literature Review. Sage.
- Hart, C. (2018). Doing a Literature Review: Releasing the Research Imagination (2nd ed.). Sage.
- Ridley, D. (2012). The Literature Review: A Step-by-Step Guide for Students (2nd ed.). Sage.
- Bohem, B. W. (1981). Software Engineering Economics. Prentice Hall.
- Chen, M., Chiang, R. H. L., & Storey, V. C. (2012). Business intelligence and analytics: From big data to big impact. MIS Quarterly, 36(4), 1165-1188.
- Han, J., Kamber, M., & Pei, J. (2012). Data Mining: Concepts and Techniques. Morgan Kaufmann.
- Kendall, K. E., & Kendall, J. E. (2011). Systems Analysis and Design. Pearson.
- Pressman, R. S. (2014). Software Engineering: A Practitioner's Approach. McGraw-Hill.
- Sommerville, I. (2011). Software Engineering. Addison-Wesley.
- Witten, I. H., Frank, E., & Hall, M. A. (2016). Data Mining: Practical Machine Learning Tools and Techniques. Morgan Kaufmann.
- Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., ... & Thomas, D. (2001). Manifesto for Agile Software Development. Agile Alliance.
- Anderson, D. J. (2010). Kanban: Successful Evolutionary Change for Your Technology Business. Blue Hole Press.
- Schwaber, K., & Sutherland, J. (2017). The Scrum Guide: The Definitive Guide to Scrum: The Rules of the Game. Scrum.org.
- Cockburn, A. (2001). Writing Effective Use Cases. Addison-Wesley Professional.
- Pender, K. (2017). The Complete Guide to UML: A Comprehensive Overview of the Unified Modeling Language. Apress.

- Booch, G., Rumbaugh, J., & Jacobson, I. (2017). The Unified Modeling Language User Guide (3rd ed.). Addison-Wesley.
- Harrison, R. (2016). UML 2.0 in Action: A Model-Driven Approach to Software Design. Springer.
- Arlow, J., & Neustadt, I. (2017). UML 2 and the Unified Process: Practical Object-Oriented Analysis and Design (2nd ed.). Addison-Wesley.
- Larman, C. (2017). Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development (6th ed.). Pearson.
- Lang, J. (2019). UML 2.0: A Developer's Guide (3rd ed.). Apress.
- IBM Corporation. (2018). UML 2.0 Tutorial. IBM Redbooks.
- Rumbaugh, J., Jacobson, I., & Booch, G. (2017). The Unified Modeling Language Reference Manual (3rd ed.). Addison-Wesley.
- Chodorow, K., & Dirolf, M. (2019). MongoDB: The Definitive Guide: Powerful and Scalable Data Storage. O'Reilly Media.
- Hodges, J. (2021). Mastering Mongoose for MongoDB. Packt Publishing.
- Myers, G. J., Sandler, C., & Badgett, T. (2011). The Art of Software Testing. John Wiley & Sons.
- Pressman, R. S. (2014). Software Engineering: A Practitioner's Approach. McGraw-Hill Education.
- Black, R. (2009). Advanced Software Testing - Vol. 1: Guide to the ISTQB Advanced Certification as an Advanced Test Analyst. Rocky Nook, Inc.
- WeddingWire. (n.d.). About WeddingWire. Retrieved from https://www.weddingwire.com

APPENDICES

Appendix : Interview Transcripts

This section includes the transcripts of interviews conducted with

General Questions

Question: What are the main challenges you face with the current wedding venue booking process?

- **Answer:** The main challenges include limited online visibility, a cumbersome manual booking process, and difficulty in managing venue availability. Many users also struggle with finding venues that match their specific needs.

Question: How does your organization currently handle venue bookings and management?

- **Answer:** We use a combination of spreadsheets and a basic booking software. While this setup works, it lacks real-time updates and integration with our website, leading to occasional discrepancies and delays.

Question: What features would be most beneficial in an online booking system for your organization?

- **Answer:** Features like real-time availability updates, automated reminders for bookings, and a user-friendly interface for both clients and staff would be highly beneficial. Integration with payment systems also add value.

Question: How important is it for you to see detailed information about a venue online before making a booking?

- **Answer:** It’s very important. Detailed information helps me make an informed decision and avoid wasting time visiting venues that don’t meet our needs.

Question: How do you typically find and compare wedding venues?

- **Answer:** I usually search online, check reviews, and visit a few venues in person. It would be great if this process could be streamlined online.

Question: How important is it for you to have the option to pay online when booking a venue?

- **Answer:** It's very important. Online payment makes the process quicker and more secure, and it also simplifies keeping track of transactions.

Question: Do you prefer booking venues directly online or through a third-party service? Why?

- **Answer:** I prefer booking directly online because it's usually more straightforward and doesn't involve additional fees.

Question: How do you currently handle booking inquiries and reservations?

- **Answer:** We handle bookings through email and phone calls, which can lead to miscommunications and delays in updating availability.

Data Security and Privacy

Question: How concerned are you about the security of your personal information when using an online system?

- **Answer:** "I am quite concerned. I would expect the system to have strong security measures, such as encryption, to protect my personal information from unauthorized access."

Question: What measures would you expect an online system to have in place to protect your data?

- **Answer:** "I would expect the system to have secure login procedures, data encryption, regular security audits, and a clear privacy policy explaining how my data will be used and protected."