HOUSEBOAT BOOKING SYSTEM

A Project Report Submitted in Partial Fulfilment of the

Requirements for the Award of the Degree of

BACHELOR OF COMPUTER APPLICATIONS

By

Anto Evaniyose - 210021089362

Under the guidance of

Ms. Leema George

Assistant Professor

Department of Computer Science

SANTHIGIRI

COLLEGE OF COMPUTER SCIENCES

Affiliated to MG University and Approved by AICTE

MARCH 2024

SANTHIGIRI

COLLEGE OF COMPUTER SCIENCES

Affiliated to MG University and Approved by AICTE

CERTIFICATE

This is to certify that the report titled ***Houseboat Booking System*** *is a bonafide record of work done by* ***Anto Evaniyose (210021089362)*** *of* ***Santhigiri College of Computer Sciences*** *in partial fulfillment of the requirements of* ***Sixth*** *Semester of* ***Bachelor of Computer Applications*** *during the* ***year 2024****.*

Rev.Fr. Prof. Dr. Baby Joseph CMI	**Mr. Gibin George**	**Ms. Leema George**
Principal	**HOD**	**Project Guide**

Internal Examiner	**External Examnier**

DECLARATION

I **Anto Evaniyose** hereby declare that the project report, titled **“Houseboat Booking System”** is a record of original work undertaken by us for the award of the degree of Bachelor of Computer Applications. I have completed this project under the guidance of Ms. Leema George, Assistant Professor, Department of Computer Science.

I also declare that this project has not been submitted for the award of any degree. I hereby confirm the originality of the work.

Name of the Student	**:**
Register Number	**:**
Signature	**:**

Date **:**

Place :

ACKNOWLEDGEMENT

A project is not complete if one fails to acknowledge all who have been instrumental in the successful completion of the project. If words were to be the symbol of undiluted feelings and token of gratitude, then let the words play the heralding role of expressing our gratitude.

First of all, I thank the "**God Almighty**" for his immense grace and blessings in our life and at each stage of this project.

I express my sincere and profound gratitude to **Rev.Fr. Prof. Dr. Baby Joseph CMI,** Principal, Santhigiri College of Computer Sciences, Vazhithala for providing all the facilities during the period of the project.

I extend my gratitude to **Mr. Gibin George** , Head of the Department of Computer Science, who is a constant source of inspiration and whose advice helped us to complete this project successfully.

I express my deep sense of gratitude to my internal project guide, **Ms. Leema George,** Assistant Professor, Department of Computer Science, for her profound guidance for the successful completion of this project.

We are grateful to all the member of **SANTHISOFT TECHNOLOGIES** for their kind support and suggestion during all the development stages.

With great enthusiasm I express my gratitude to all the faculty members of Department of Computer Science for their timely help and support.

Finally, I express our deep appreciation to all my friends and family members for the moral support and encouragement they have given to complete this project successfully.

ABSTRACT

The proposed web application for houseboats aims to seamlessly connect three key stakeholders: administrators, owners, and users. The system will empower administrators with robust tools to efficiently manage and oversee the entire platform, ensuring smooth operations and user interactions. Owners will benefit from a comprehensive dashboard, enabling them to list their houseboats, update availability, and monitor reservations. Users, seeking unique and personalized accommodation experiences, will access a user-friendly interface to browse listings, make reservations, and engage in secure transactions.

The project scope encompasses user authentication, dynamic listing creation, reservation management, payment processing, and an intuitive administrative interface. By addressing the specific needs of each stakeholder, this web application will enhance the overall experience of managing and booking houseboat accommodations.

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

IDE	Integrated Development Environment
CPU	Central Processing Unit
DBMS	Data Base Management System
RDBMS	Relational Data Base Management System
NF	Normal Forms
PK	Primary Key
FK	Foreign Key
DFD	Data Flow Diagram

TABLE OF CONTENTS

1. INTRODUCTION

1.1 BACKGROUND AND MOTIVATION

The concept of launching an online platform for houseboat bookings stems from the fusion of two timeless desires – the allure of waterfront living and the ease of modern technology. As urbanization progresses, the quest for unique and immersive travel experiences has intensified, leading to a growing fascination with unconventional accommodations. Houseboats, nestled in serene waters, epitomize a harmonious blend of luxury and nature, offering an escape from the mundane. The motivation behind the venture is rooted in the belief that such floating abodes provide an unparalleled sense of tranquility and scenic beauty, capturing the essence of slow travel. By harnessing the power of the internet, we aim to simplify the booking process, making these extraordinary stays accessible to a global audience. The online platform not only caters to the wanderlust of travelers seeking distinctive lodgings but also aligns with the evolving preferences for sustainable and eco-friendly tourism. This initiative envisions fostering a community of houseboat enthusiasts, connecting them with idyllic destinations worldwide. As the demand for experiential travel continues to rise, the houseboat online booking platform strives to be a gateway to a world where the gentle lapping of water against the hull becomes the soundtrack to unforgettable journeys. In essence, the venture encapsulates the spirit of adventure, the charm of unconventional accommodations, and the convenience of digital booking, paving the way for an immersive travel experience that transcends the ordinary.

1.2 THE PROPOSED SYSTEM

The proposed system for houseboat online booking aims to streamline the process of renting houseboats by integrating a user-friendly online platform, with key stakeholders being the admin, owners, and users. The admin will have centralized control, managing overall system functionality, user accounts, and ensuring security. Owners of houseboats will have dedicated profiles, allowing them to list their properties, set pricing, and update availability. Users, seeking a unique lodging experience, will access the platform to browse, compare, and book houseboats seamlessly. The system will feature an intuitive interface, providing users with detailed information about each houseboat, including amenities, location, and reviews. The booking process will be secure and efficient, with users able to make reservations, track their bookings, and communicate with owners. For owners, the system offers a convenient way to showcase their houseboats to a broader audience, manage reservations, and receive payments. Admin will play a crucial role in ensuring fair practices, resolving disputes, and maintaining the integrity of the platform. Overall, the proposed system facilitates a transparent and efficient online marketplace for houseboat rentals, benefiting owners by expanding their reach and providing users with a hassle-free booking experience.

1.3 PROJECT SCOPE

1.3.1. Limitations of Existing System

- The more the number of houseboats, the greater the amount of paperwork for the company, requiring significant manual effort.
- The current system allows users to search for houseboat rentals through traditional methods such as printed advertisements, posters, newspapers, and visual media like websites. However, this process is time-consuming.
- Finding the right houseboat that matches specific preferences, such as amenities and rental rates, poses a challenge for users. Additionally, attending houseboat fairs, if available, may require attendees to submit paper-based booking requests.

1.3.2. Advantages of Proposed System

- The admin can efficiently manage and monitor the entire booking process through a centralized system. This reduces the chances of errors, improves data accuracy, and streamlines administrative tasks.
- Owners benefit from increased occupancy rates as the online booking system provides a platform for users to easily discover and reserve houseboats. This can lead to better revenue generation and optimized utilization of resources.
- Owners receive real-time updates on bookings, cancellations, and other relevant information. This enables them to make timely decisions, manage resources effectively, and respond promptly to user inquiries.
- Users experience enhanced convenience with the ability to browse, compare, and book houseboats online. The system provides a user-friendly interface, simplifying the booking process and saving time for users.

- Users benefit from transparent communication regarding booking details, pricing, and availability. This reduces uncertainties, enhances trust, and ensures a positive user experience.
- Automation of routine processes, such as booking confirmations, payment processing, and availability updates, reduces manual workload for both the admin and owners. This results in increased efficiency and decreased chances of errors.
- The system ensures the security of sensitive data, including user information and financial transactions. This instills confidence in all stakeholders and promotes the safe use of the online booking platform.

2. SYSTEM ANALYSIS

2.1 INTRODUCTION

Software Engineering is the analysis, design, construction, verification and management of technical or social entities. To engineer software accurately, a software engineering process must be defined. System analysis is a detailed study of the various operations performed by the system and their relationship within and module of the system. It is a structured method for solving the problems related to the development of a new system. The detailed investigation of the present system is the focal point of system analysis. This phase involves the study of parent system and identification of system objectives. Information has to be collected from all people who are affected by or who use the system. During analysis, data are collected on the variable files, decision point and transactions handled by the present system. The main aim of system is to provide the efficient and user friendly automation. So the system analysis process should be performed with extreme precision, so that an accurate picture of existing system, its disadvantages and the requirements of the new system can be obtained.

System analysis involves gathering the necessary information and using the structured tool for analysis. This includes the studying existing system and its drawback, designing a new system and conducting cost benefit analysis. System analysis is a problem solving activity that requires intensive communication between the system users and system developers. The system is studied to the minute detail and analyzed. The system is viewed as a whole and the inputs to the system are identified. The outputs from the organization are traced through various phases of processing of inputs.

There are a number of different approaches to system analysis. When a computer based information system is developed, systems analysis (according to the Waterfall model) would constitute the following steps:

- The development of a feasibility study, involving determining whether a project is economically, technologically and operationally feasible.

- Conducting fact-finding measures, designed to ascertain the requirements of the system's end-users. These typically span interviews, questionnaires, or visual observations of work on the existing system

- Gauging how the end-users would operate the system (in terms of general experience in using computer hardware or software), what the system would be used for and so on.

Techniques such as interviews, questionnaires etc. can be used for the detailed study of these processes. The data collected by these sources must be scrutinized to arrive at a conclusion.

The conclusion is an understanding of how the system functions. This system is called the Existing System. The Existing system is then subjected to close observation and the problem areas are identified. The designer now functions as a problem solver and tries to sort out the difficulties that the enterprise faces. The solutions are given as a proposal which is the Proposed System. The proposal is then weighed with the existing system analytically and the best one is selected. The proposal is then presented to the user for an endorsement by the user. The proposal is reviewed on user request and suitable changes are made. This is a loop that ends as soon as the user is satisfied with the proposal.

2.2. STAKE HOLDERS OF THIS PROJECT

2.2.1. Administrator

The administrator plays a crucial role in ensuring a seamless and secure booking process, addressing customer inquiries, and maintaining the integrity of the platform. Additionally, financial stakeholders such as payment processors and the administrator's own organization depend on efficient transactions. Local communities and regulatory bodies may also be stakeholders, as the administrator must navigate compliance with relevant laws and regulations. Balancing the interests of these stakeholders is essential for the administrator to foster trust, sustain a thriving marketplace, and contribute to the overall success of the houseboat booking platform.

2.2.2. Owner

The customers who book houseboats through the platform are crucial stakeholders, as their satisfaction directly impacts the success of the business. Ensuring a user-friendly interface, transparent booking processes, and excellent customer service is vital to retain and attract clients. Additionally, the houseboat owners who list their properties on the platform are essential stakeholders. The owner must balance their needs by providing effective tools and support for property management, ensuring fair revenue-sharing models, and maintaining positive relationships. Furthermore, the platform must consider regulatory bodies and local authorities to comply with laws governing the houseboat rental industry. A harmonious relationship with all stakeholders is key to the sustainable growth and success of the online booking platform for houseboats.

2.2.3. User

Users are interested in a seamless booking experience, comprehensive information about available houseboats, transparent pricing, and a user-friendly interface. They expect the platform to provide accurate details about amenities, location, and customer reviews to make informed decisions. Additionally, users anticipate secure payment options and responsive customer support for any queries or concerns. As the central figure in the houseboat booking process, meeting user expectations and ensuring a positive overall experience is crucial for the success and sustainability of the online platform.

2.3. SOFTWARE REQUIREMENT SPECIFICATION

2.3.1. Admin

1. Admin should have the facility to view their own homepage.
2. Admin should have the facility to view owners registeration request.
3. Admin should have the facility to accept or deny registeration of owners.
4. Admin should have the facility to view users registeration request.
5. Admin should have the facility to accept or deny registeration of users.
6. Admin should have the facility to add/view/edit/delete locations.
7. Admin should have the facility to add/view/edit/delete boat types.

8. Admin should have the facility to add/view/edit/delete packages.
9. Admin should have the facility to add/view/edit/delete services.
10. Admin should have the facility to add/view/edit/delete boats.
11. Admin should have the facility to view boats.
12. Admin should have the facility to view payment.
13. Admin should have the facility to view booking details.
14. Admin should have the facility to view owners.
15. Admin should have the facility to view reports.

2.3.2. Owner

1. Owner should have the facility to view their own homepage.
2. Owner should have the facility to add service package.
3. Owner should have the facility to booking status.
4. Owner should have the facility to view boat types.

2.3.3. User/Customer

1. User should have the facility to view their own homepage.
2. User should have the facility to view boat types.
3. User should have the facility to book the boats.
4. User should have the facility to schedule date and time.
5. User should have the facility to settle the payment.

Table 2.1. Sign off table

Sl. No.	Name & Designation	Date	Accepted (Yes/No)
1	Ms. Leema George Assistant Professor Santhigiri College of Computer Sciences		
2	Anto Evaniyose Developer		

2.4. FEASIBILITY STUDY

Feasibility is defined as the practical extent to which a project can be performed successfully. To evaluate feasibility, a feasibility study is performed, which determines whether the solution considered to accomplish the requirements is practical and workable in the software. Information such as resource availability, cost estimation for software development, benefits of the software to the organization after it is developed and cost to be incurred on its maintenance are considered during the feasibility study. The objective of the feasibility study is to establish the reasons for developing the software that is acceptable to users, adaptable to change and conformable to established standards. Various other objectives of feasibility study are listed below.

- To analyse whether the software will meet organizational requirements.

- To determine whether the software can be implemented using the current technology and within the specified budget and schedule.

- To determine whether the software can be integrated with other existing software. When my project guide as well as my client Mr. Gibin George told me regarding the main project and about Question Paper Generator for getting the desired product developed, it comes up with rough idea about what all functions the software must perform and which all features are expected from the software.

Referencing to this information, I do studies and discussions about whether the desired system and its functionality are feasible to develop and the output of this phase is a feasibility study report that should contained adequate comments and recommendations.

Various types of feasibility that we checked include technical feasibility, operational feasibility, and economic feasibility.

Technical Feasibility

Technical feasibility assesses the current resources (such as hardware and software) and technology, which are required to accomplish user requirements in the software within the allocated time and budget. For this, the software development team ascertains whether the current resources and technology can be upgraded or added in the software to accomplish specified user requirements. Technical feasibility also performs the following tasks.

- Analyses the technical skills and capabilities of the software development team members.
- Determines whether the relevant technology is stable and established.
- Ascertains that the technology chosen for software development has a large number of users so that they can be consulted when problems arise or improvements are required.

From my perspective there are two languages ASP.NET, HTML and database MySQL which are used to develop this web based applications. ASP.NET is used in the front end and MySQL is used in the back end. The Question Paper Generator is web based and thus can be accessed through any browsers. As we are using these latest technologies which are currently trending and used by a number of developers across the globe, we can say that our project is technically feasible.

Operational Feasibility

Operational feasibility assesses the extent to which the required software performs a series of steps to solve business problems and user requirements. This feasibility is dependent on human resources (software development team) and involves visualizing whether the software will operate after it is developed and be operative once it is installed. Operational feasibility also performs the following tasks.

- Determines whether the problems anticipated in user requirements are of high priority.
- Determines whether the solution suggested by the software development team is acceptable.
- Analyses whether users will adapt to a new software.
- Determines whether the organization is satisfied by the alternative solutions proposed by the software development team.

I found that my project will be satisfied for the client since I was discussing every detail about the software with the client at every step. The most important part of operational feasibility study is the input from client. So the software is built completely according to the requirements of the client. I have used the current industry standards for the software. Hence we can say that this software is operationally feasible.

Economic Feasibility

Economic feasibility determines whether the required software is capable of generating financial gains for an organization. It involves the cost incurred on the software development team, estimated cost of hardware and software, cost of performing feasibility study, and so on. For this, it is essential to consider expenses made on purchases (such as hardware purchase) and activities required to carry out software development. In addition, it is necessary to consider the benefits that can be achieved by developing the software. Software is said to be economically feasible if it focuses on the issues listed below.

- Cost incurred on software development to produce long-term gains for an organization.
- Cost required to conduct full software investigation (such as requirements elicitation and requirements analysis).
- Cost of hardware, software, development team, and training.

It is estimated that my project is economically feasible as development cost is very minimal since the tools and technologies used are available online. Development time is well planned and will not affect other operations and activities of the individuals. Once the system has been developed, the companies purchasing the system will be providing with a manual for training purposes. There is no need to purchase new hardware since the existing computers can still be used to implement the new system.

2.5. SOFTWARE DEVELOPMENT LIFECYCLE MODEL

One of the basic notions of the software development process is SDLC models which stand for Software Development Life Cycle models. SDLC – is a continuous process, which starts from the moment, when it's made a decision to launch the project, and it ends at the moment of its full remove from the exploitation. Software development lifecycle (SDLC) is a framework that defines the steps involved in the development of software. It covers the detailed plan for building, deploying and maintaining the software. SDLC defines the complete cycle of development i.e. all the tasks involved in gathering a requirement for the maintenance of a Product.

Some of the common SDLC models are Waterfall Model, V-Shaped Model, Prototype Model, Spiral Model, Iterative Incremental Model, Big Bang Model, Agile Model. We used Agile Model for our Project.

Agile Model

Agile Model is a combination of the Iterative and incremental model. This model focuses more on flexibility while developing a product rather than on the requirement. In the agile methodology after every development iteration, the client is able to see the result and understand if he is satisfied with it or he is not. Extreme programming is one of the practical use of the agile model. The basis of this model consists of short meetings where we can review our project. In Agile, a product is broken into small incremental builds. It is not developed as a complete product in one go. At the end of each sprint, the project guide verifies the product and after his approval, it is finalised. Client feedback is taken for improvement and his suggestions and enhancement are worked on in the next sprint. Testing is done in each sprint to minimize the risk of any failures.

Advantages of Agile Model:

- It allows more flexibility to adapt to the changes.
- The new feature can be added easily.
- Customer satisfaction as the feedback and suggestions are taken at every stage.
- Risks are minimized thanks to the flexible change process.

Disadvantages:

- Lack of documentation.
- If a customer is not clear about how exactly they want the product to be, then the project would fail.
- With all the corrections and changes there is possibility that the project will exceed expected time

2.6. HARDWARE AND SOFTWARE REQUIREMENTS.

2.6.1. Software Specification

This project is built upon the latest technology software.

Technology	:	PYTHON DJANGO
Front end	:	HTML, CSS, JAVASCRIPT
Back end	:	PYTHON
Framework	:	Django, Bootstrap
Database	:	MYSQL
Web Server	:	WAMP
Operating System	:	Windows 11

2.6.1.1 PYTHON DJANGO

Django, an open-source web framework written in Python, facilitates rapid and scalable web development with a clean, pragmatic design. At its core is the Model-ViewTemplate (MVT) architectural pattern, comprising models for data structures, views for handling user requests and rendering responses, and templates for defining HTML structure. Django's Object-Relational Mapping (ORM) simplifies database interaction by translating Python models into corresponding database tables. The framework boasts a dynamic admin interface for effortless database management and customization. URL routing is declarative, providing a clear structure for web applications, while the template system allows the separation of HTML and Python code. Middleware components enhance request/response processing, enabling tasks like authentication and security checks. Django includes a form handling system for easy validation and processing of HTML forms, and its security features guard against common vulnerabilities such as Cross-Site Scripting (XSS) and Cross-Site Request Forgery (CSRF). The built-in authentication system manages user-related functionalities, while the testing framework ensures code reliability. Optionally, the Django REST framework extends capabilities for building robust Web APIs with features like serializers, authentication, and view sets. With a command-line interface (CLI) for various tasks, Django empowers developers to create scalable, secure, and maintainable web applications efficiently, making it a preferred choice for web development projects globally.

2.6.1.2 MySQL

MySQL is the world's most popular open-source database software, with over 100 million copies of its software downloaded or distributed throughout its history. With its superior speed, reliability, and ease of use, MySQL has become the preferred choice for Web, Web 2.0, SaaS, ISV, Telecom companies and forward-thinking corporate IT Managers because it eliminates the major problems associated with downtime, maintenance and administration for modern, online applications.

Many of the world's largest and fastest-growing organizations use MySQL to save time and money powering their high-volume Web sites, critical business systems, and packaged software — including industry leaders such as Yahoo!, Alcatel-Lucent, Google, Nokia, YouTube, Wikipedia, and Booking.com.

The flagship MySQL offering is MySQL Enterprise, a comprehensive set of production tested software, proactive monitoring tools, and premium support services available in an affordable annual subscription.

MySQL is a key part of LAMP (Linux, Apache, MySQL, PHP / Perl / Python), the fastgrowing open-source enterprise software stack. More and more companies are using LAMP as an alternative to expensive proprietary software stacks because of its lower cost and freedom from platform lock-in.

MySQL was originally founded and developed in Sweden by two Swedes and a Finn: David Axmark, Allan Larsson and Michael "Monty" Widenius, who had worked together since the 1980's. MySQL, the most popular Open-Source SQL database management system, is developed, distributed, and supported by Oracle Corporation.

MySQL is a database management system. A database is a structured collection of data. It may be anything from a simple shopping list to a picture gallery or the vast amounts of information in a corporate network. To add, access, and process data stored in a computer database, you need a database management system such as MySQL Server. Since computers are very good at handling large amounts of data, database management systems play a central role in computing, as standalone utilities, or as parts of other applications.

MySQL databases are relational. A relational database stores data in separate tables rather than putting all the data in one big storeroom. The database structures are organized into physical files optimized for speed. The logical model, with objects such as databases, tables, views, rows, and columns, offers a flexible programming environment. You set up rules governing the relationships between different data fields, such as one-to one, one-to many, unique, required or optional, and ―pointers‖ between different tables. The database enforces these rules, so that with a well-designed database, your application never sees inconsistent, duplicate, orphan, out-of-date, or missing data.

The SQL part of ―MySQL‖ stands for ―Structured Query Language‖. SQL is the most common standardized language used to access databases. Depending on your programming environment, you might enter SQL directly (for example, to generate reports), embed SQL statements into code written in another language, or use a language specific API that hides the SQL syntax.

SQL is defined by the ANSI/ISO SQL Standard. The SQL standard has been evolving since 1986 and several versions exist. In this manual, ―SQL-92‖ refers to the standard released in 1992, ―SQL:1999‖ refers to the standard released in 1999, and ―SQL:2003‖ refers to the current version of the standard.

We use the phrase ―the SQL standard‖ to mean the current version of the SQL Standard at any time. MySQL software is Open Source. Open-Source means that it is possible for anyone to use and modify the software. Anybody can download the MySQL software from the Internet and use it without paying anything. If you wish, you may study the source code and change it to suit your needs.

The MySQL software use the GPL (GNU General Public License), http://www.fsf.org/licenses/, to define what you may and may not do with the software in different situations. If you feel uncomfortable with the GPL need to embed MySQL code into a commercial application, you can buy a commercially licensed version from us.

The MySQL Database Server is very fast, reliable, scalable, and easy to use. If that is what you are looking for, you should give it a try. MySQL Server can run comfortably on a desktop or laptop, alongside your other applications, web servers, and so on, requiring little or no attention. If you dedicate an entire machine to MySQL, you can adjust the settings to take advantage of all the memory, CPU power, and I/O capacity available. MySQL can also scale up to clusters of machines, networked together.

MySQL Server was originally developed to handle large databases much faster than existing solutions and has been successfully used in highly demanding production environments for several years. Although under constant development, MySQL Server today offers a rich and useful set of functions. Its connectivity, speed, and security make MySQL Server highly suited for accessing databases on the Internet. MySQL Server works in client/server or embedded systems.

The MySQL Database Software is a client/server system that consists of a multi- threaded SQL server that supports different backends, several different client programs and libraries, administrative tools, and a wide range of application programming interfaces (APIs). We also provide MySQL Server as an embedded multi-threaded library that you can link into your application to get a smaller, faster, easier-to-manage standalone product.

A large amount of contributed MySQL software is available. MySQL Server has a practical set of features developed in close cooperation with our users. It is very likely that your favourite application or language supports the MySQL Database Server.

2.6.1.3 WAMP SERVER

WAMP Server is a Windows web development environment. It allows you to create web applications with Apache2, PHP and a MySQL database. Alongside, PhpMyAdmin allows you to manage easily your databases. WAMP Server refers to a software stack for the Microsoft Windows operating system, created by Romain Bourdon and consisting of the Apache web server, Open SSL for SSL support, MySQL database and PHP programming language. WAMP Server is a Web development platform on Windows that allows you to create dynamic Web applications with Apache2, PHP, MySQL and MariaDB. WampServer automatically installs everything you need to intuitively developed Web applications. You will be able to tune your server without even touching its setting files. Best of all, WampServer is available for free (under GPML license) in both 32and64 bit versions. WampServer is not compatible with Windows XP, SP3, or Windows Server 2003.

WAMP Server's functionalities are very complete and easy to use so we won't explain here how to use them.

With a left click on WAMP Server's icon, you will be able to:

- manage your Apache and MySQL services
- switch online/offline (give access to everyone or only localhost)

- install and switch Apache, MySQL and PHP releases
- manage your server's settings
- access your logs
- access your settings files
- create alias

2.6.1.4 PyCharm Community

PyCharm Community is a free, open-source integrated development environment (IDE) designed specifically for Python development. Developed by JetBrains, PyCharm Community provides a comprehensive set of features tailored to meet the needs of Python developers. The IDE offers a clean and intuitive user interface with powerful coding assistance, including intelligent code completion, code analysis, and quick fixes, enhancing productivity and reducing development time. PyCharm Community supports various Python frameworks, including Django, Flask, and Pyramid, providing project templates and specific tools for web development. It seamlessly integrates with version control systems like Git, facilitating collaborative development. The built-in visual debugger allows for efficient debugging of Python code, with features such as breakpoints, watches, and a visual console. PyCharm Community also supports test-driven development (TDD) with integrated testing tools for running and debugging tests. The platform comes equipped with a powerful SQL database tool, enabling developers to manage databases directly within the IDE. Additionally, PyCharm Community supports popular web technologies, such as HTML, CSS, and JavaScript, providing a full-stack development experience. Its compatibility with various build tools, virtual environments, and package managers enhances flexibility. The IDE fosters a dynamic development environment with realtime code analysis and error highlighting, promoting code quality and readability. The extensive plugin ecosystem further extends functionality, allowing developers to customize and enhance their IDE experience. PyCharm Community's commitment to user experience is evident through regular updates, addressing performance improvements, bug fixes, and the incorporation of new features. With its robust set of tools, extensive language support, and active community engagement, PyCharm Community remains a go-to choice for Python developers seeking a powerful, free IDE to streamline their coding workflows and enhance the overall development experience.

2.6.1.5 WINDOWS 10

Operating System is defined as a program that manages the computer hardware. An operating system can be viewed as a scheduler, where it has resources for which it has charge. Resources include CPU, memory, I/O device and disk space. In another view, the operating system is a new machine. The third view is that operating system is a multiplexer which allows sharing of resources provides protection from interference and provides a level of cooperation between users. This project is developed using Windows 10 as the operating system and supports its latest versions. Windows 10 is a series of personal computer operating systems produced by Microsoft as part of its Windows NT family of operating systems. It is the successor to Windows 8.1, and was released to manufacturing on July 15, 2015, and to retail on July 29, 2015. One of Windows 10's most notable features is support for universal apps. Windows 10 also introduced the Microsoft Edge web browser, a virtual desktop system, a window and desktop management feature called Task View, support for fingerprint and face recognition login, new security features for enterprise environments, and DirectX12. Windows 10 received mostly positive reviews upon its original release in July 2015. Critics praised Microsoft's decision to provide a desktop-oriented interfacing line with previous versions of Windows, contrasting the tablet-oriented approach of 8, although Windows 10's touch-oriented user interface mode was criticized for containing regressions upon the touch-oriented interface of Windows 8. Critics also praised the improvements to Windows 10's bundled software over Windows 8.1, Xbox Live integration, as well as the functionality and capabilities of the Cortana personal assistant and the replacement of Internet Explorer with Microsoft Edge. However, media outlets have been critical of changes to operating system behaviours, including mandatory update installation, privacy concerns over data collection performed by the OS for Microsoft and its partners and the adware-like tactics used to promote the operating system on its release.

2.6.1.6 MICROSOFT WORD

Microsoft Word (or simply Word) is a word processor developed by Microsoft. It was first released on October 25, 1983 under the name *Multi-Tool Word* for Xenix systems. Subsequent versions were later written for several other platforms including IBM PCs running DOS (1983), Apple Macintosh running the Classic Mac OS (1985), AT&T Unix PC (1985), Atari ST (1988), OS/2 (1989), Microsoft Windows (1989), SCO Unix (1994), and macOS (formerly OS X; 2001).

Commercial versions of Word are licensed as a standalone product or as a component of Microsoft Office, Windows RT or the discontinued Microsoft Works suite. Unlike most MS-DOS programs at the time, Microsoft Word was designed to be used with a mouse. Advertisements depicted the Microsoft Mouse, and described Word as a WYSIWYG, windowed word processor with the ability to undo and display bold, italic, and underlined text, although it could not render fonts. It was not initially popular, since its user interface was different from the leading word processor at the time, WordStar. However, Microsoft steadily improved the product, releasing versions 2.0 through 5.0 over the next six years. In 1985, Microsoft ported Word to the classic Mac OS (known as Macintosh System Software at the time). This was made easier by Word for DOS having been designed for use with high-resolution displays and laser printers, even though none were yet available to the general public. Following the precedents of Lisa Write and MacWrite, Word for Mac OS added true WYSIWYG features. It fulfilled a need for a word processor that was more capable than MacWrite. After its release, Word for Mac OS's sales were higher than its MS-DOS counterpart for at least four years.

2.6.1.7 SMARTDRAW

Smart Draw is a diagram tool used to make flowcharts, organization charts, mind maps, project charts, and other business visuals. Smart Draw has two versions: an online edition and a downloadable edition for Windows desktop.

Smart Draw integrates with Microsoft Office products including Word, PowerPoint, and Excel and G Suite applications like Google Docs and Google Sheets. Smart Draw has apps for Atlassian's Confluence, Jira, and Trello. Smart Draw is compatible with Google Drive, Dropbox, Box, and OneDrive.

Since 1994, the mission of Smart Draw Software has been to expand the ways in which people communicate so that we can clearly understand each other, make informed decisions, and work together to improve our businesses and the world. We accomplish this by creating software and services that make it possible for people to capture and present information as visuals, while being a pleasure to use. In 2019, we took this to the next level by launching Visual Script, which makes it easy to visualize data in relational formats like trees, flows, and timelines, automatically, without any human input. Visual Script is a relationship visualization platform that empowers organizations to visualize data across siloed ecosystems and gain critical insights in real-time. Today, Smart Draw Software is one of the most sophisticated digital marketing organizations in the world with over 90,000 unique visitors to our website each business day and in excess of 3,000,000 installations of our apps each year. Smart Draw is used by more than half of the Fortune 500 and by over 250,000 public and private enterprises of all sizes around the world. Privately held, Smart Draw Software is headquartered in San Diego, California.

2.6.2 HARDWARE REQUIREMENTS

The selection of hardware configuring is a very task related to the software development, particularly inefficient RAM may affect adversely on the speed and corresponding on the efficiency of the entire system. The processor should be powerful to handle all the operations. The hard disk should have the sufficient to solve the database and the application.

Hardware used for development:

CPU	: Intel i5 Processer
Memory	: 4 GB
Cache	: 6 MB
Hard Disk	: 1 TB
Monitor	: 15.6” Monitor
Keyboard	: Standard108 keys Enhanced Keyboard
Mouse	: Optical Mouse

Minimum Hardware Required for Implementation:

CPU	: Pentium IV Processor
Memory	: 256MB Above
Cache	: 512 KB Above
Hard Disk	: 20 GB Above
Monitor	: Any
Keyboard	: Any
Mouse	: Any

3.SYSTEM DESIGN

3.1 SYSTEM ARCHITECTURE

A system architecture or system's architecture is the conceptual model that defines the structure, behavior and more views of a system. An architecture description is a formal description and representation of a system, organized in a way that supports reasoning about the structures of the system,

System architecture can comprise system components, the externally visible properties of those components, the relationships (e.g the behavior) between them. It can provide a plan from which products can be procured, and systems developed, that will work together to implement the overall system. There have been efforts to formalize languages to describe system architecture; collectively these are called architecture description languages (ADLs).

The system architecture can best be thought of as a set of representations of an existing (or to be created) system. It is used to convey the informational content of the elements comprising a system, the relationships among those elements, and the rules governing those relationships. The architectural components and set of relationships between these components that architecture describes may consist of hardware, software, documentation, facilities, manual procedures, or roles played by organizations or people. System architecture is primarily concerned with the internal interfaces among the system's components or subsystems, and the interface between the system and its external environment, especially the user.

The structural design reduces complexity, facilitates change and result in easier implementation by encouraging parallel development of different parts of the system. The procedural design transforms structural elements of program architecture into a procedural description of software components. The architectural design considers architecture as the most important functional requirement. The system is based on the three-tier architecture.

The first level is the user interface (presentation logic), which displays controls, receives and validates user input. The second level is the business layer (business logic) where the application specific logic takes place. The third level is the data layer where the application information is stored in files or database. It contains logic about to retrieve and update data. The important feature about the three-tier design is that information only travels from one level to an adjacent level.

3.2. MODULE DESIGN

Modular programming is a software design technique that emphasizes separating the functionality of a program into independent, interchangeable modules, such that each contains everything necessary to execute only one aspect of the desired functionality. Conceptually, modules represent a separation of concerns, and improve maintainability by enforcing logical boundaries between components. Different modules of this project includes

1. **Authentication**

   This module allows the admin and owner to log into our website. Admin and owner can log into the system by using their corresponding username and password. They can perform all actions in the system only after login. Users/customers log into the system using their corresponding username and password. After logging in, users/customers can access the details and connect to the system.

2. **Registeration**

   This module encompasses the entire registration process within the system, catering to various stakeholders including Admin, Owner and User/Customer. Admin privileges extend to Location registration,category registration, package registration, service registration. Owner, who has specific registration tasks related to boat ownership or management. The registration process encompasses sub-modules like Boat registration, user/customer view payment request and accept or deny the payment. User/Customers are required to register on the website, providing essential information such as Name, Address, phone number, email id, and password. This data is stored in the database, enabling Admin to utilize it for various purposes. The Admin holds the authority to approve or reject user registrations. Once registered, users can access and view pertinent information on the site. This comprehensive registration system ensures a structured and secure environment for all stakeholders involved.

3. **Activities**

   This module is designed to facilitate seamless interaction among three stakeholders like Admin, User/Customer, and Owner in the context of houseboat bookings. Users can easily explore various houseboat types, along with their respective prices, on the website. Upon selecting a preferred houseboat, users can proceed to book it by specifying the desired time and date. The booking process concludes with the determination of a net amount payable by the customer. Once a booking is made, the Owner plays a pivotal role in approving or rejecting the order. Upon approval, the customer is prompted to make the payment through online channels such as Google Pay or Paytm. In addition to the booking functionality, the system also incorporates a mechanism for users to raise doubts and share their houseboat experiences through reviews. The Admin is equipped to address customer doubts, enhancing the system's credibility and user satisfaction. Furthermore, the module introduces an Owner role, who has a comprehensive view of their houseboat listings, bookings, and customer reviews. This multifaceted approach aims to attract users by ensuring a transparent and efficient houseboat booking system, benefitting both customers and owners.

4. **Reports**

This module allows the admin to view the details of the site through various reports. The reports provide the valuable information from the system. In our system all reports are important because the system eliminate the use of manipulation of the paper work. The system generates useful information's. Admin can generate many reports such that list of houseboat, list of boat category and list of customers etc... So that the Admin can easily manage the website. Admin can also generate the report include the list of customers who made more bookings, list of the houseboat which is mostly booked, list of locations most bookings done and also these can be represent in a report. This will helpful for the admin to generate useful information. The reports show the significance of the system.

3.3. DATABASE DESIGN

A database is a collection of interrelated data stored with minimum redundancy to serve many users quickly and efficiently. The general objective is to make information access easy, quick, inexpensive and flexible for the users. The general theme behind a database is to integrate all information. Database design is recognized as a standard of management information system and is available virtually for every computer system. In database design several specific objectives are considered:

- Ease of learning and use
- Controlled redundancy

- Data independence
- More information at low cost
- Accuracy and integrity
- Recovery from failure
- Privacy and security
- Performance

A database is an integrated collection of data and provides centralized access to the data. Usually the centralized data managing the software is called RDBMS. The main significant difference between RDBMS and other DBMS is the separation of data as seen by the program and data has in direct access to stores device. This is the difference between logical and physical data.

3.3.1 Normalization

Designing a database is complete task and the normalization theory is a useful aid in the design process. The process of normalization is concerned with transformation of conceptual schema into computer representation form. There will be need for most databases to grow by adding new attributes and new relations. The data will be used in new ways. Tuples will be added and deleted. Information stored may undergo updating also. New association may also be added.
In such situations the performance of a database is entirely depend upon its design.

A bad database design may lead to certain undesirable things like:

- Repetition of information
- Inability to represent certain information
- Loss of information

To minimize these anomalies, Normalization may be used. If the database is in a normalized form, the data can be growing without, in most cases, forcing the rewriting application programs. This is important because of the excessive and growing cost of maintaining an organization's application programs and its data from the disrupting effects of database growth. As the quality of application programs increases, the cost of maintaining the without normalization will rise to prohibitive levels. A normalized database can also encompass many related activities of an organization thereby minimizing the need for rewriting the applications of programs. Thus, normalization helps one attain a good database design and there by ensures continued efficiency of database.

Normalization theory is built around the concept of normal forms. A relation is said to be in normal form if it satisfies a certain specified set of constraints. For example, a relation is said to be in first normal form (1NF) if it satisfies the constraint that it contains atomic values only. Thus every normalized relation is in 1NF.Numerous normal forms have been defined.

Codd defined the first three normal forms.

All normalized relations are in 1NF, some 1NF relations are also in 2NF and some 2NF relations are also in 3NF.2NF relations are more desirable than 1Nf and 3NF are more desirable than 2NF. That is, the database designer should prefer 3NF than 1NF or 2NF.Normalization procedure states that a relation that is in some given normal form can be converted into a set of relations in a more desirable form. We can define this procedure as the successive reduction of a given collection of relations to some more desirable form. This procedure is reversible. That is, it is always possible to take the output from the procedure and convert them back into input. In this process, no information is lost. So it is also called "no loss decomposition".

First Normal Form

A relation is in first normal form (1NF) if and all its attributes are based on single domain. The objective of normalizing a table is to remove its repeating groups and ensure that all entries of the resulting table have at most single value.

Second Normal Form

A table is said to be second Normal Form (2NF), when it is in 1NF and every attribute in record is functionally dependent upon the whole key, and not just a part of the key.

Third Normal Form

A table is in third Normal Form (3NF), when it is in 2NF and every non-key attribute is functionally dependent on just the primary key.

3.3.2 Table Structure

Table is a collection of complete details about a particular subject. These data are saved in rows and Columns. The data of each Row are different units. Hence, rows are called RECORDS and Columns of each row are called FIELDS.
Data is stored in tables, which is available in the backend the items and data, which are entered in the input, form id directly stored in this table using linking of database. We can link more than one table to input forms. We can collect the details from the different tables to display on the output.

There are mainly 11 tables in the project. They are,

1. tbl_login
2. tbl_district
3. tbl_location
4. tbl_category
5. tbl_package
6. tbl_service
7. tbl_owner
8. tbl_boat
9. tbl_customer

10. tbl_book

11. tbl_payment

1. Table: **tbl_login**

Description: To store login details.

Table 3.1 tbl_login

Field name	Data type	Constraints	Description
loginid	Int(11)	Primary key	To store uniquely identify each stakeholder's and the values are automatically generate
username	Varchar(50)	Not null	To store the username of the stakeholder
password	Varchar(50)	Not null	To store the password of the stakeholder
role	Varchar(50)	Not null	This field is used to store role of each stakeholder
status	Varchar(50)	Not null	This fields identify the log on status of each stakeholder

2. Table: **tbl_district**

Description: To store district details.

Table 3.2 tbl_district

Field name	Data type	Constraints	Description
districtid	Int(11)	Primary key	Field used to uniquely identify districts and the values are generated automatically
districtname	Varchar(50)	Not null	To store the district name

3. Table: **tbl_location**

Description: To store location details.

Table 3.3 tbl_location

Field name	**Data type**	**Constraints**	**Description**
locationid	Int(11)	Primary key	Field used to uniquely identify locations and the values are generated automatically
locationname	Varchar(20)	Not null	To store the location name
districtid	Int(11)	Foreign key	This field is used to take the reference from tbl_district

4. Table: **tbl_category**

Description: To describe the category of the boat.

Table 3.4 tbl_category

Fieldname	**Data type**	**Constrains**	**Discription**
categoryid	Int	Primary key	Field used to uniquely identify category and the values are generated automatically
categoryname	Varchar(20)	Not null	To store category name
image	Varchar(100)	Not null	To store category image

5. Table: **tbl_package**

Description: To store the package details.

Table 3.5 tbl_package

Field name	**Data type**	**Constraints**	**Description**
packageid	Int	Primary key	Field used to uniquely identify package and the values are generated automatically
packagename	Varchar(50)	Not null	To store package name
description	Varchar(50)	Not null	To store package details
amount	Int	Not null	To store the total amount of the package.
days	Int	Not null	To store the day selected by the user from the start to the end of the journey

6.Table: **tbl_service**

Description: To store the sevice details.

Table 3.6 tbl_service

Field name	**Data type**	**Constraints**	**Description**
serviceid	Int	Primary key	Field used to uniquely identify service and the values are generated automatically
servicename	Int	Not null	To store service name
amount	Int	Not null	To store the total amount of the service.

7.Table: **tbl_owner**

Description: To store owner details.

Table 3.7 tbl_owner

Field name	**Data type**	**Constraints**	**Description**
ownerid	Int	Primary key	Field used to uniquely identify owner and the values are generated automatically
ownername	Varchar(50)	Not null	To store owner name
locationid	Int	Foreign key	This field is used to take the reference from tbl_location
email	Varchar(50)	Not null	To store the email id of owner
contactno	Bigint(20)	Not null	To store contact number of owner
regdate	Date	Not null	To store the registration date of the owner
loginid	Int	Foreign key	This field is used to take the reference from tbl_login

8.Table: **tbl_boat**

Description: To store boat details.

Table 3.8 tbl_boat

Field name	**Data type**	**Constraints**	**Description**
boatid	Int	Primary key	Field used to uniquely identify boat and the values are generated automatically
packageid	Int	Foreign key	This field is used to take the reference from tbl_package
serviceid	Int	Foreign key	This field is used to take the reference from tbl_service
regdate	Date	Not null	To store the registration date of the boat
ownerid	Int	Foreign key	This field is used to take the reference from tbl_owner
image	Varchar(100)	Not null	This field is for uploading image of a boat.
boattype	Varchar(50)	Not null	This field is for representing the boat type.

9.Table: **tbl_customer**

Description: To store customer details.

Table 3.9 tbl_customer

Field name	**Data type**	**Constraints**	**Description**
customerid	Int	Primary key	Field used to uniquely identify customer and the values are generated automatically
customername	Varchar(50)	Not null	To store customer name
housename	Varchar(50)	Not null	To store house name
locationid	Int	Foreign key	This field is used to take the reference from tbl_location
pincode	Bigint(20)	Not null	To store pin code of customer
emailid	Varchar(50)	Not null	To store emailid of customer
phoneno	Bigint(20)	Not null	To store phone number of customer
regdate	Date	Not null	To store the registration date of the customer
loginid	Int	Foreign key	This field is used to take the reference from tbl_login

10.Table: **tbl_book**

Description: To store booking details.

Table 3.10 tbl_book

Field name	**Data type**	**Constraints**	**Description**
bookid	Int	Primary key	Field used to uniquely identify booking details and the values are generated automatically
boatid	Int	Foreign key	This field is used to take the reference from tbl_boat
customerid	Int	Foreign key	This field is used to take the reference from tbl_customer
requestdate	Date	Not null	To store request date
status	Varchar(50)	Not null	This field is used to know that the booking status is successful or failed.
requireddate	Date	Not null	To store required date

remark	Varchar(50)	Not null	This field is used for owner response to user after payment request.
days	Int	Not null	To store number of days
persons	Int	Not null	To store number of persons

11.Table: **tbl_payment**

Description: To store payment details.

Table 3.11 tbl_payment

Field name	**Data type**	**Constraints**	**Description**
paymentid	Int	Primary key	Field used to uniquely identify payment details and the values are generated automatically
bookid	Int	Foreign key	This field is used to take the reference from tbl_book
customerid	Int	Foreign key	This field is used to take the reference from tbl_customer
requestdate	Date	Not null	To store request date
status	Varchar(50)	Not null	This field is used to know that the payment status is successful or failed.

3.3.3 Data Flow Diagram

3.3.3.1 Introduction to Data Flow Diagrams

Data Flow Diagram is a network that describes the flow of data and processes that change, or transform, data throughout the system. This network is constructed by use a set of symbols that do not imply a physical implementation. It is a graphical tool for structured analysis of the system requirements. DFD models a system by using external entities from which data flows to a process, which transforms the data and creates, output-data-flows which go to other processes or external entities or files. Data in files may also flow to processes as inputs. There are various symbols used in a DFD. Bubbles represent the processes. Named arrows indicate the data flow. External entities are represented by rectangles. Entities supplying data are known as sources and those that consume data are called sinks. Data are stored in a data store by a process in the system. Each component in a DFD is labelled with a descriptive name. Process names are further identified with a number.

The Data Flow Diagram shows the logical flow of a system and defines the boundaries of the system. For a candidate system, it describes the input (source), outputs (destination), database (files) and procedures (data flow), all in a format that meet the user’s requirements.

The main merit of DFD is that it can provide an overview of system requirements, what data a system would process, what transformations of data are done, what files are used, and where the results flow.

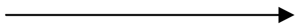
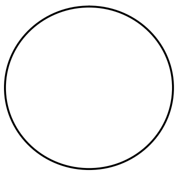
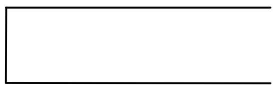
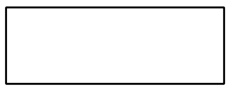
This network is constructed by use a set of symbols that do not imply a physical implementation. It is a graphical tool for structured analysis of the system requirements. DFD models a system by using external entities from which data flows to a process, which transforms the data and creates, output-data-flows which go to other processes or external entities or files. External entities are represented by rectangles. Entities supplying data are known as sources and those that consume data are called sinks. Data are stored in a data store by a process in the system. It is a graphical tool for structured analysis of the system requirements. DFD models a system by using external entities from which data flows to a process, which transforms the data and creates, output-data-flows which go to other processes or external entities or files. Data in files may also flow to processes as inputs.

Rules for constructing a Data Flow Diagram

1. Arrows should not cross each other
2. Squares, circles and files must bear names.
3. Decomposed data flow squares and circles can have same time
4. Choose meaningful names for data flow
5. Draw all data flows around the outside of the diagram

Basic Data Flow Diagram Symbols

Table 3.12 Data Flow Diagram Symbols

	A data flow is a route, which enables packets of data to travel from one point to another. Data may flow from a source to a process and from data store or process. An arrow line depicts the flow, with arrow head pointing in the direction of the flow.
	Circles stands for process that converts data in to information. A process represents transformation where incoming data flows are changed into outgoing data flows.
	A data store is a repository of data that is to be stored for use by a one or more process may be as simple as buffer or queue or sophisticated as relational database. They should have clear names. If a process merely uses the content of store and does not alter it, the arrowhead goes only from the store to the process. If a process alters the details in the store, then a double-headed arrow is used.
	A source or sink is a person or part of an organization, which enters or receives information from the system, but is considered to be outside the contest of data flow model.

3.3.3.2 Data Flow Diagram

Each component in a DFD is labelled with a descriptive name. Process name are further identified with number. Context level DFD is draw first. Then the process is decomposed into several elementary levels and is represented in the order of importance. A DFD describes what data flow (logical) rather than how they are processed, so it does not depend on hardware, software, and data structure or file organization.

A DFD methodology is quite effective; especially when the required design.

Zeroth level DFD for Houseboat

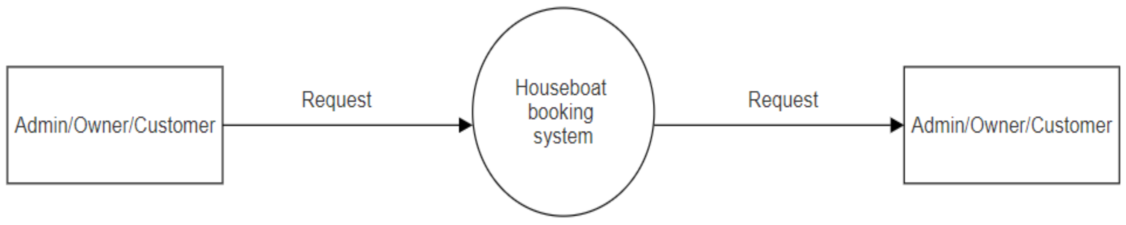


Fig 3.1 Context level DFD for Houseboat Booking system

First level DFD for Houseboat Booking system

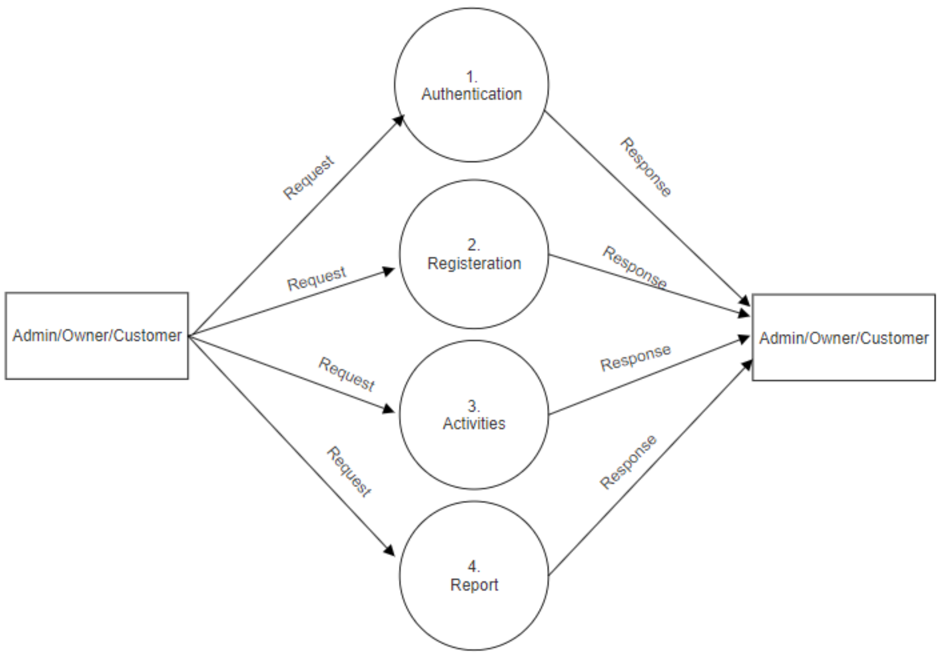


Fig 3.2 First level DFD for Houseboat Booking system

Second level DFD for User Authentications

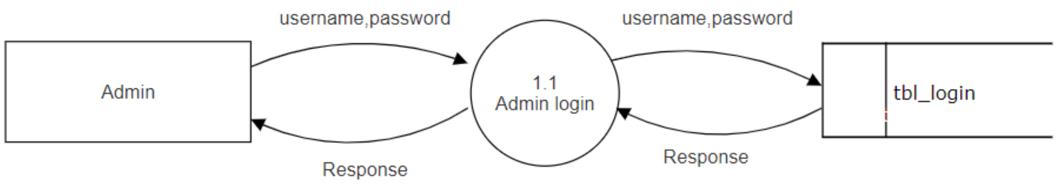


Fig. 3.3 Second level DFD for Admin login

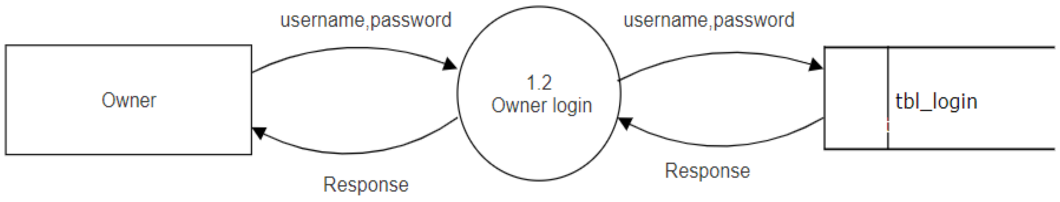


Fig. 3.4 Second level DFD for Owner login

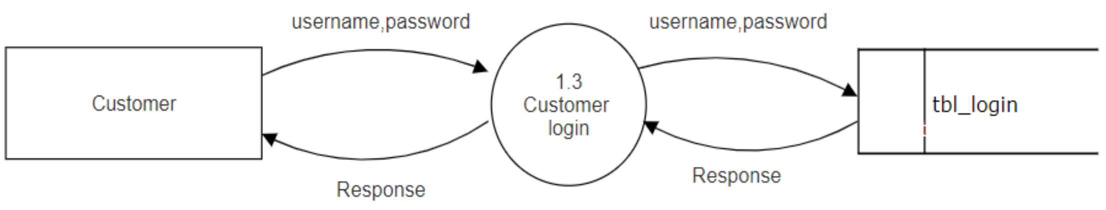


Fig. 3.5 Second level DFD for Customer login

Second level DFD for Registration

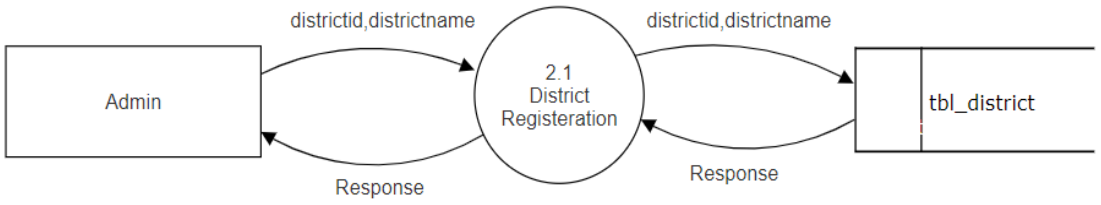


Fig. 3.6 Second level DFD for District registeration

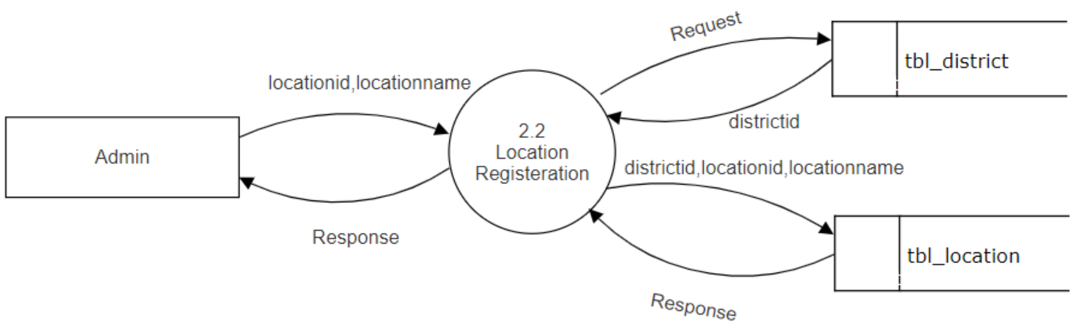


Fig. 3.7 Second level DFD for Location registeration

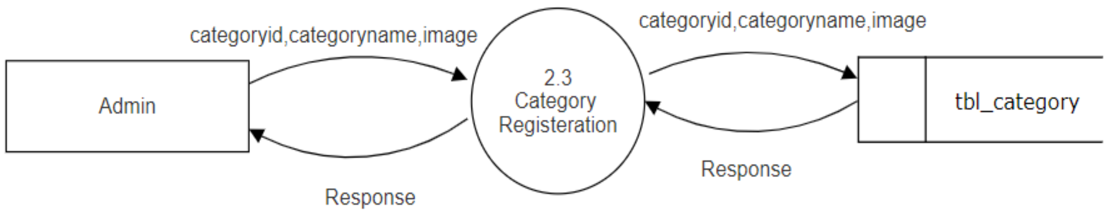


Fig. 3.8 Second level DFD for Category Registeration

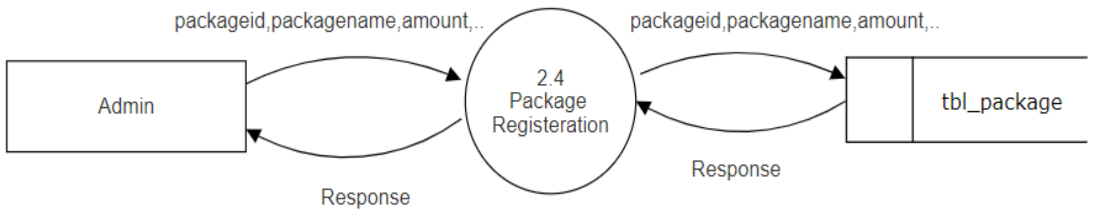

packageid,packagename,amount,..
Admin
Response
2.4
Package
Registeration
packageid,packagename,amount,..
Response
tbl_package

Fig. 3.9 Second level DFD for Package registeration

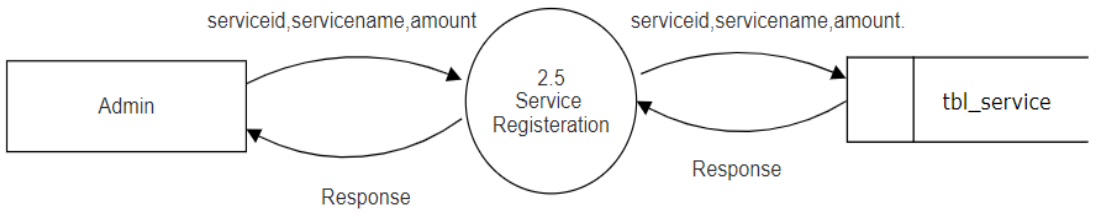


Fig. 3.10 Second level DFD for Service registeration

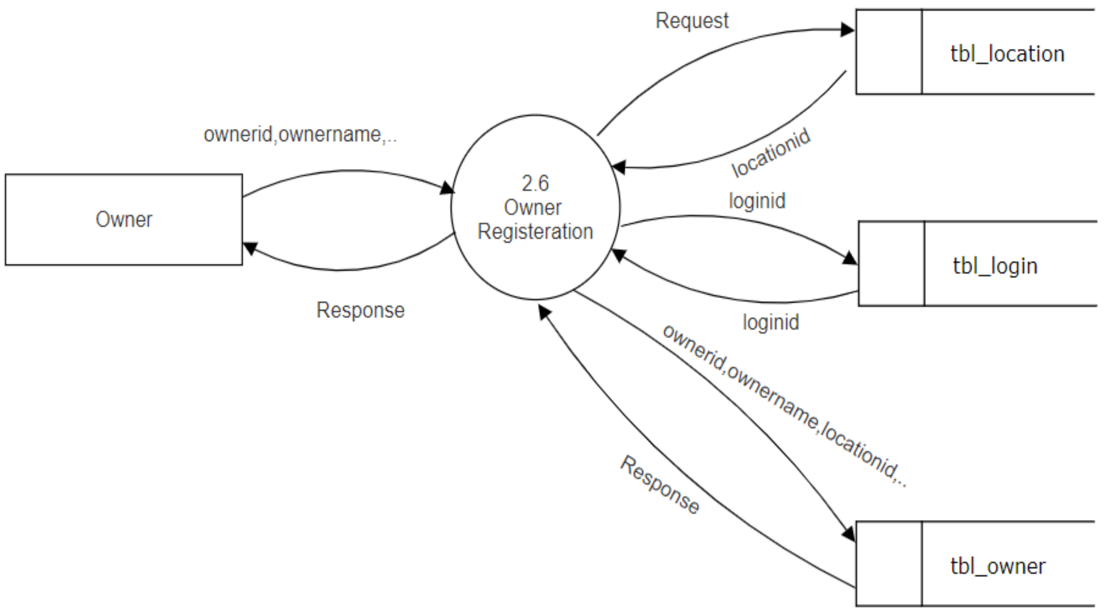


Fig. 3.11 Second level DFD for Owner registeration

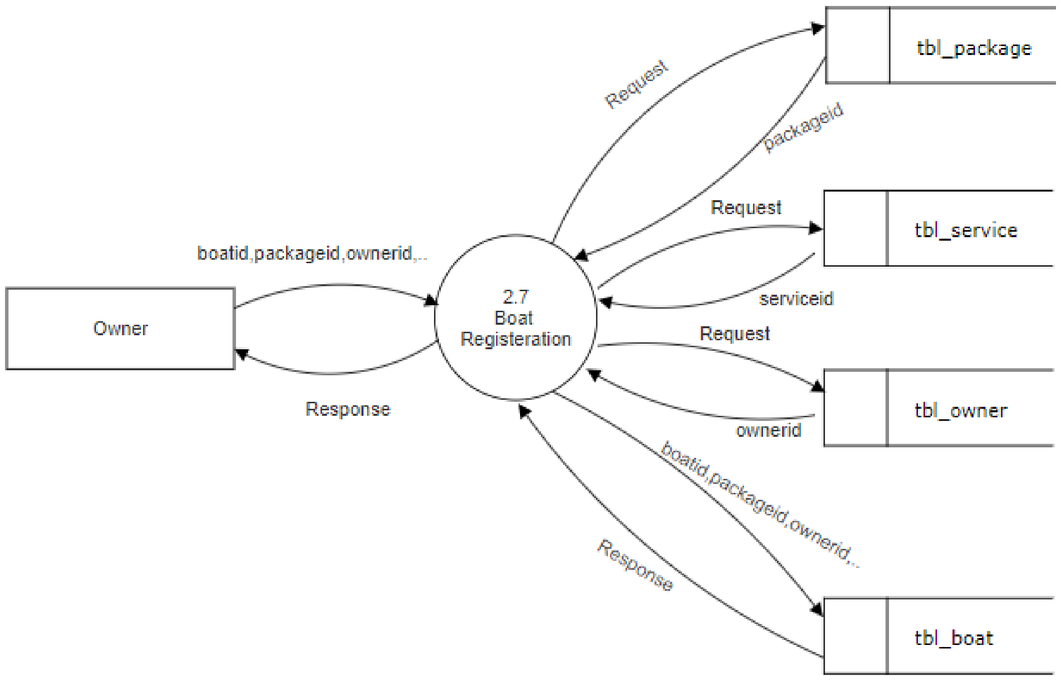


Fig. 3.12 Second level DFD for Boat registeration

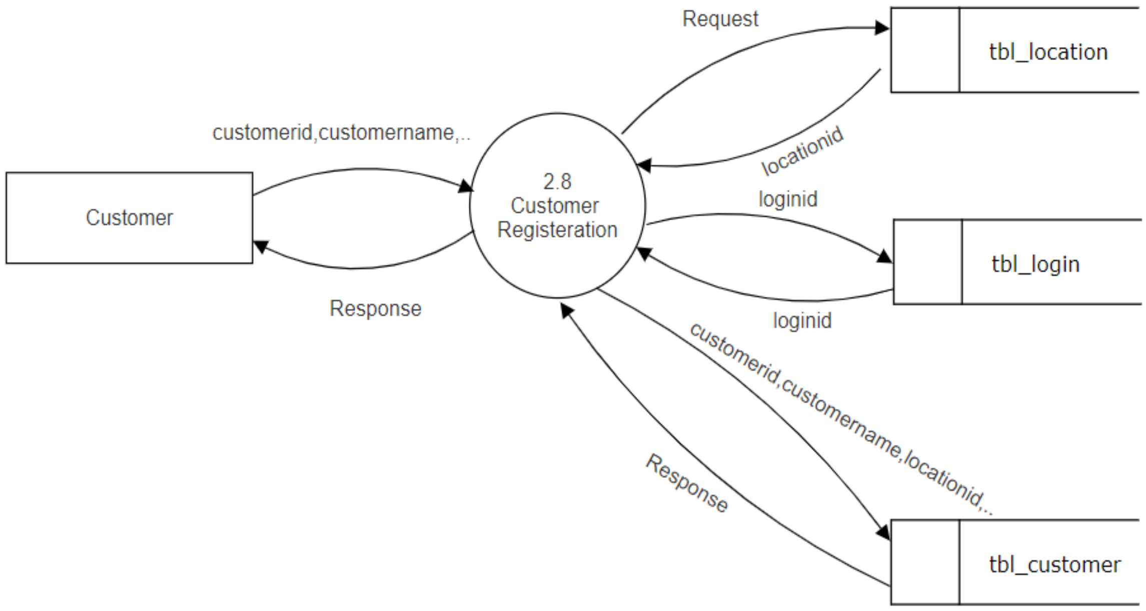


Fig. 3.13 Second level DFD for Customer registeration

Second level DFD for Activities

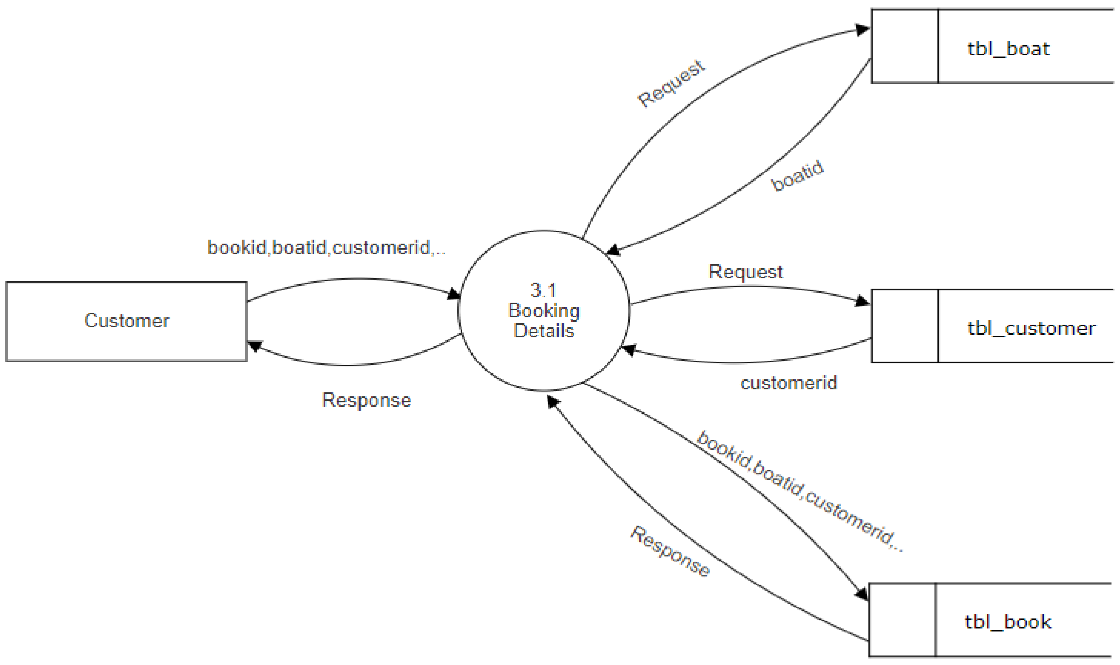


Fig. 3.14 Second level DFD for booking details

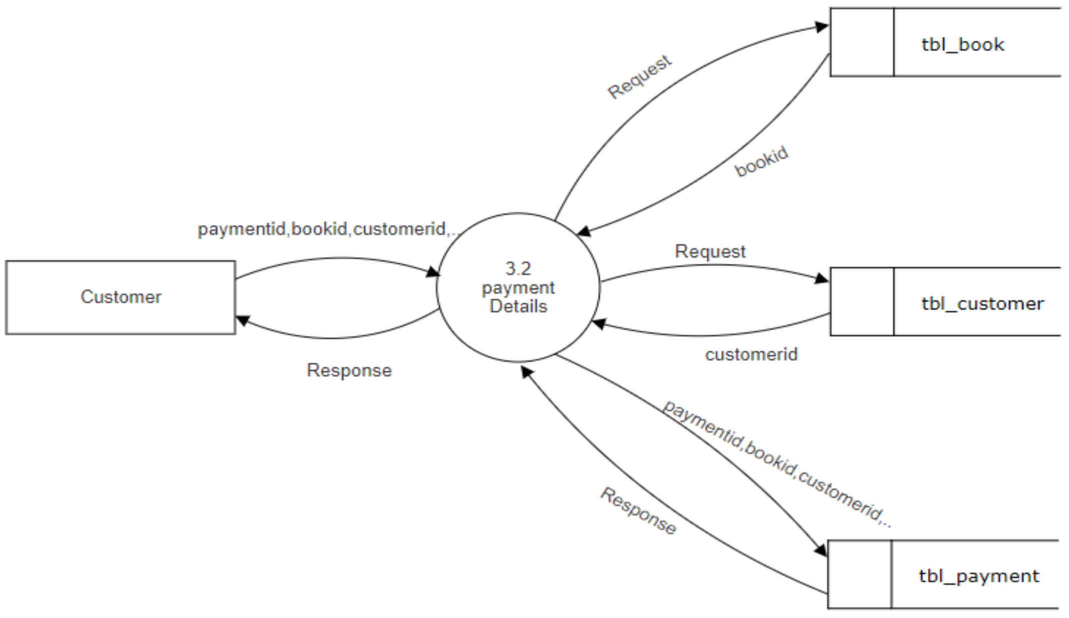


Fig. 3.15 Second level DFD for payment details

3.4 INTERFACE DESIGN

These modules can apply to hardware, software or the interface between a user and a machine. An example of a user interface could include a GUI, a control panel for a nuclear power plant, or even the cockpit of an aircraft. In systems engineering, all the inputs and outputs of a system, subsystem, and its components are listed in an interface control document often as part of the requirements of the engineering project. The development of a user interface is a unique field.

3.4.1 User Interface Screen Design

The user interface design is very important for any application. The interface design describes how the software communicates within itself, to system that interpreted with it and with humans who use it. The input design is the process of converting the useroriented inputs into the computer based format. The data is fed into the system using simple inactive forms. The forms have been supplied with messages so that the user can enter data without facing any difficulty. They data is validated wherever it requires in the project. This ensures that only the correct data have been incorporated into system. The goal of designing input data is to make the automation as easy and free from errors as possible. For providing a good input design for the application easy data input and selection features are adopted. The input design requirements such as user friendliness, consistent format and interactive dialogue for giving the right messages and help for the user at right are also considered for development for this project. Input Design is a part of the overall design. The input methods can be broadly classified into batch and online. Internal controls must be established for monitoring the number of inputs and for ensuring that the data are valid.

The basic steps involved in input design are:

• Review input requirements.

• Decide how the input data flow will be implemented.

• Decide the source document.

• Prototype on line input screens.

• Design the input screens.

The quality of the system input determines the quality of the system output. Input specifications describe the manner in which data enter the system for processing. Input design features can ensure the reliability of the system and produce results from accurate data. The input design also determines whether the user can interact efficiently with the system.

This input form is for the registration of new user. It contains textboxes for inputting Full Name, Gender, Email, Date of Join, Contact Number, Username and Password. This form allows the user to select Designation through dropdown selection. After clicking the Submit button the user will get a registration successful and the password is automatically provided to the user through mail and the user can login to our website using the email id and password. The user registration form is very important in the project. This allows the user to enter their details and register in the system before login in to the system. This helpful for the users to prove their credential. Each user must have to fill the full details that are given in the form to register into the system and log in to it. Each field have its own label that denotes the value need to enter in that box. Also each textbox have placeholders which helpful for the user to decide the type of value which need to enter in the box. The form also has a button that allows the user to pass the contents entered in the form to the database table. The data entered in the form should be correct according to the type of that field. All labels are arranged in the same alignment line and all boxes to enter values are also in the same line.

3.4.2 Output Design

A quality output is one, which meets the requirements of end user and presents the information clearly. In any system result of processing are communicated to the user and to the other system through outputs. In the output design it is determined how the information is to be displayed for immediate need. It is the most important and direct source information is to the user. Efficient and intelligent output design improves the system's relationships with the user and helps in decision -making. The objective of the output design is to convey the information of all the past activities, current status and to emphasis important events. The output generally refers to the results and information that is generated from the system. Outputs from computers are required primarily to communicate the results of processing to the users.

Output also provides a means of storage by copying the results for later reference in consultation. There is a chance that some of the end users will not actually operate the input data or information through workstations, but will see the output from the system.
Two phases of the output design are:

1. Output Definition
2. Output Specification

Output Definition takes into account the type of output contents, its frequency and its volume, the appropriate output media is determined for output. Once the media is chosen, the detail specification of output documents are carried out. The nature of output required from the proposed system is determined during logical design stage. It takes the outline of the output from the logical design and produces output as specified during the logical design phase.
In a project, when designing the output, the system analyst must accomplish the following:

• Determine the information to present.

• Decide whether to display, print, speak the information and select the output medium.

• Arrange the information in acceptable format.

• Decide how to distribute the output to the intended receipt.