UE21CS341A-SoftwareEngineering

Deliverable 2

“Online Motorcycle Bike Rental System”

“System Plan and Design Document”

Team:

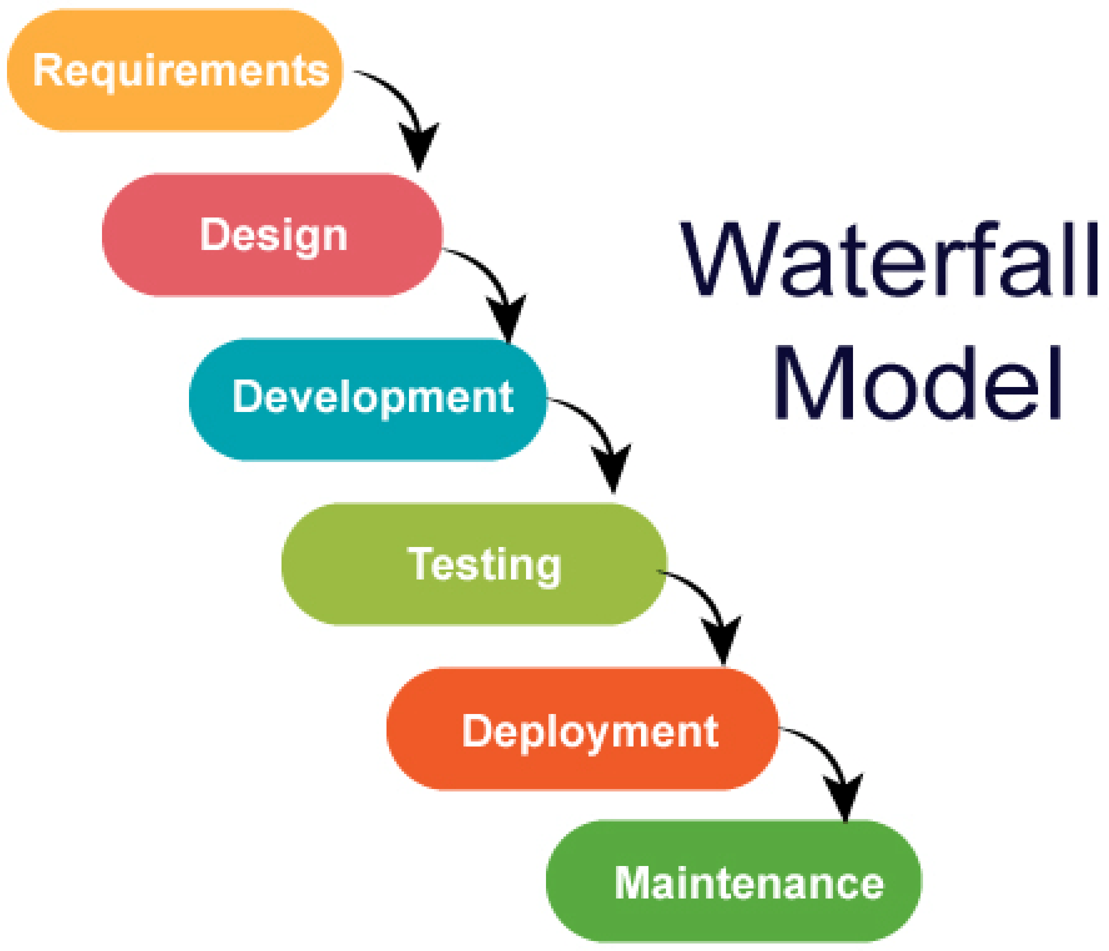
Preetham V Divakara --PES1UG22CS448

Prithvi Raj B L --PES1UG22CS451

PROJECT PLAN DOCUMENT

Life-cycle followed :

The lifecycle we are going to follow is waterfall model since the requirements for the projects are well planned and there might be no high changes to be made during the project. We intend to complete one phase before the start of another phase and also there is need for strict adherence of deadlines. Also, since waterfall modal can ensure every requirement is made properly and even documentation is ready at particular time. We tend to use agile methodologies wherever changes are required in the middle of projects.



Tools Used for this Project:

1. **PHP (Hypertext Preprocessor):**
   - Purpose: Used for building the web-based front-end and back-end of the bike rental system.
   - Usage: Developing the overall website with user and admin functionalities.
2. **MySQL:**
   - Purpose: Relational Database Management System (RDBMS).
   - Usage: Storing user data, bike details, bookings, and other relevant data for the system.
3. **GitHub:**
   - Purpose: Collaboration tool for version control and managing code repositories.
   - Usage: Project management, sharing code, and tracking version history among team members.
4. **Visual Studio Code:**
   - Purpose: Development environment for writing and testing code.
   - Usage: Front-end and back-end development using PHP, HTML, CSS, and MySQL.
5. **SSL/TLS (HTTPS):**
   - Purpose: Secure communication between the client and server.
   - Usage: Ensuring data security when users interact with the system.

Work Breakdown Structure (WBS)

1. Planning:

1.1 Project Initiation:

- Define project objectives:
    - Create a web-based platform for renting motorcycles with user and admin functionalities.

- Identify stakeholders:
    - Customers (bike renters), Administrators (managing bike listings), Developers (team members).

- Tools: Collaboration via GitHub for code sharing and team communication.

1.2 Requirements Gathering:

- Use the **SRS** document to define system features:
    - User authentication, bike search and booking, booking management, admin management.

- Prioritize core functionalities such as user authentication, bike search, and admin dashboard over secondary features like payment integration.

2. Designing:

2.1 UI/UX Design:

- Design user-friendly interfaces for customers and administrators, ensuring simplicity for users while offering detailed controls for admins.

- Tools: HTML, CSS for layout and style, integrated with PHP for dynamic content rendering.

2.2 Architecture Design:

- Develop a system architecture, involving:
    - Front-end: PHP for rendering web pages.
    - Back-end: MySQL for database management and PHP for handling requests.
    - Security: SSL/TLS for secure user sessions and transactions.

- System must be scalable to handle an increasing number of users and bikes.

3. Front-end Development:

3.1 Development Environment Setup:

- Set up a LAMP (Linux, Apache, MySQL, PHP) stack for development.

- Tool: Visual Studio Code for writing and testing PHP code, setting up the PHP server, and managing MySQL databases.

3.2 Front-end Implementation:

- Implement key pages:
    - Login/Signup page for user authentication.
    - Bike search page that allows filtering by location, brand, and availability.
    - Booking management interface.

- Integrate PHP with MySQL to fetch and display data dynamically.

4. Set up Database:

4.1 MySQL Database Setup:

- Define tables for users, bikes, and bookings based on the SRS's system features.
- Design **CRUD (Create, Read, Update, Delete)** operations for admin functionality.

- Ensure database security by encrypting sensitive data such as passwords and payment information.

5. Testing and Verifying:

5.1 Functional Testing:

- Test individual components like user registration, bike search, and booking management.
- Use sample data in the MySQL database to test search and booking functionalities.

5.2 Bug Identification and Resolution:

- Identify bugs using manual testing methods.

- Resolve issues related to user management, booking conflicts, or incorrect data display.

6. Deployment:

6.1 Deployment Setup:

- Push the project code to GitHub for version control.

- Deploy the system to a web server capable of running PHP and MySQL (e.g., using a cloud service like AWS or DigitalOcean).

- Implement SSL certificates for HTTPS support.

7. Documentation, Training, and Project Closure:

7.1 User Documentation:

- Create user manuals for customers:
    - Steps for logging in, searching for bikes, and booking rentals.

- Admin documentation:
    - Guide for managing bike listings, user accounts, and viewing booking statistics.

7.2 Training Resources:

- Provide developer training resources, including environment setup, database configuration, and integration instructions.

7.3 Project Closure:

- Final review and evaluation of project milestones.
- Archive code and documentation in GitHub for future use or modification.

8.Effort Estimation (in person-months):

- Number of team members: **2**
- Estimated project duration: **12 weeks** (3 months)
- Assumption: Each person works **6 hours/week**.

Total Person-hours = 2 (members) * 6 (hours/week) * 12 (weeks)

= **144 person-hours**

Converting to person-months:

Average monthly working hours = 4.33(weeks/month) * 5 (days/week) * 8(hours/day)

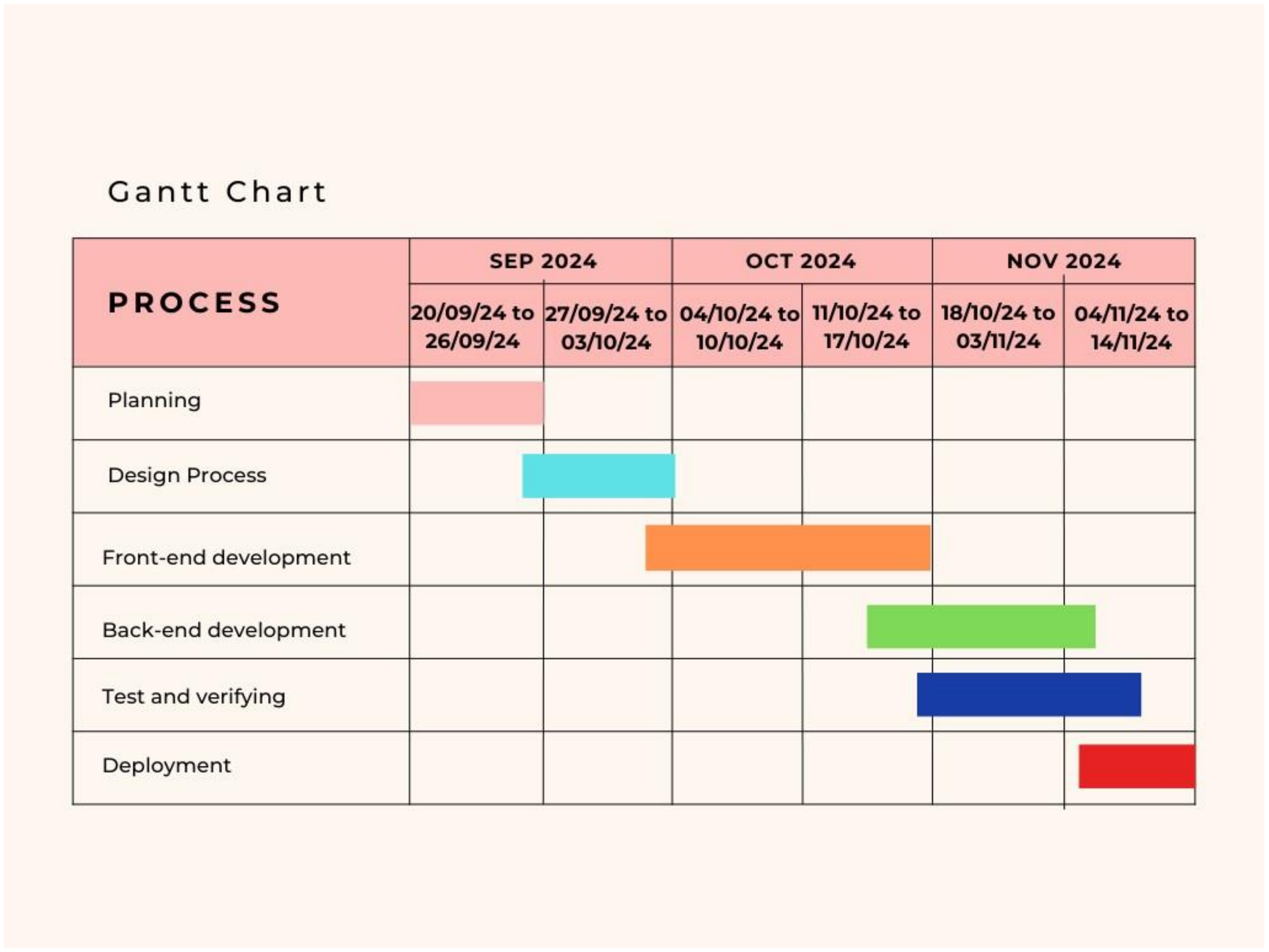
= **173.2 hours/month**

Total person-months = 144 hours /(173.2 hours/month) = **0.83 person-months**

Effort per person:

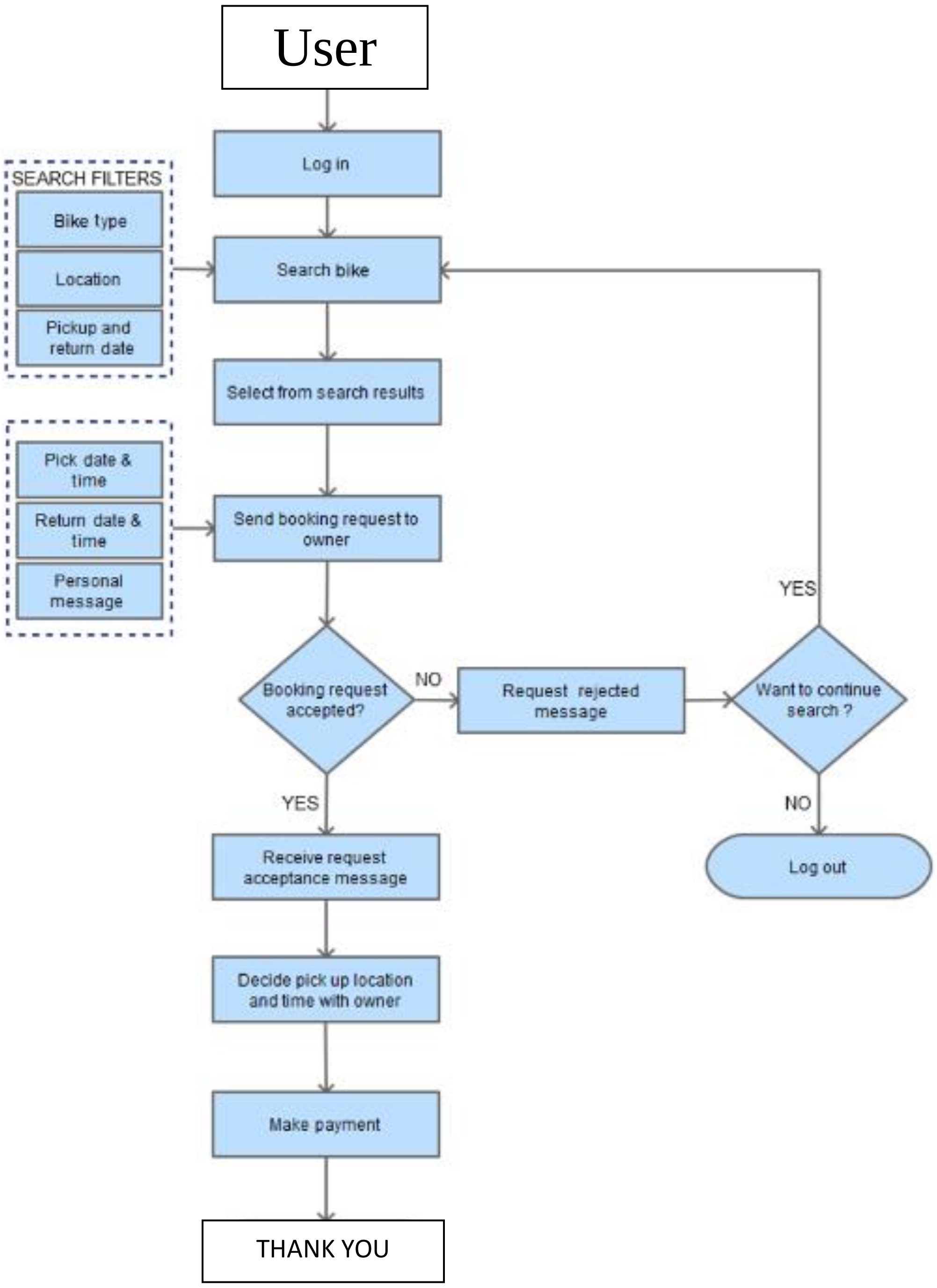
- **Total effort per person** = 0.83 / 2 = **0.415 person-months** per person.

PROJECT DEVELOPMENT PROCESS

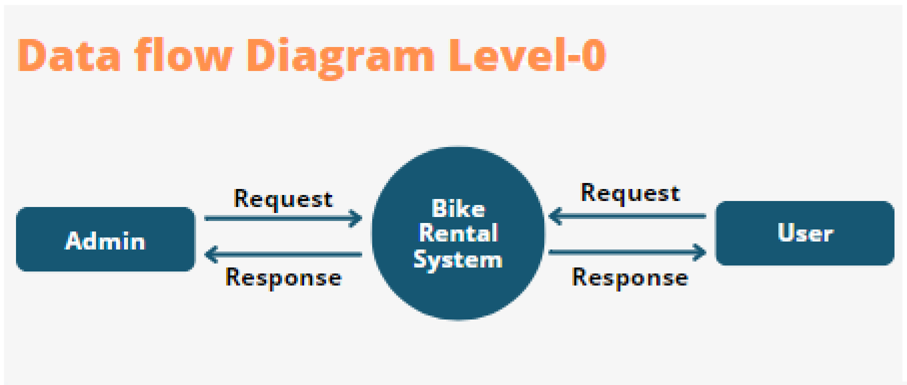


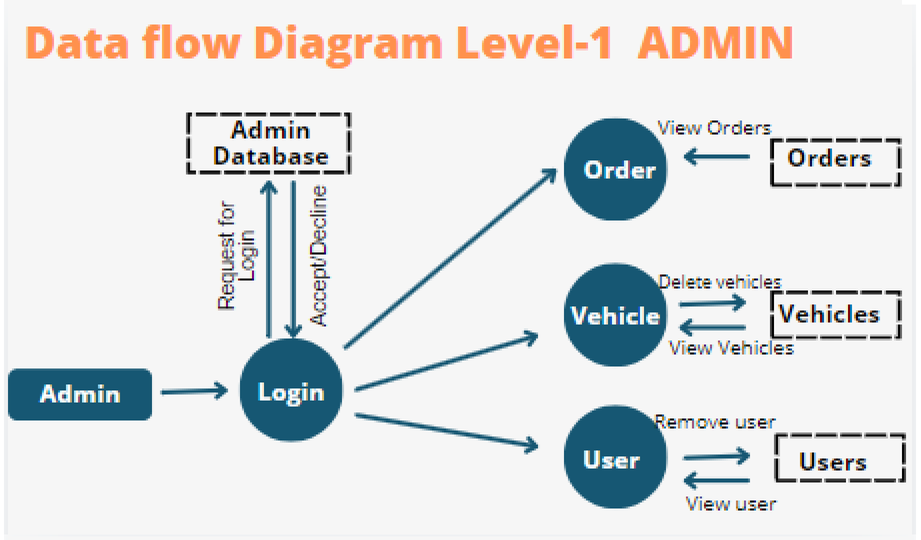
DESIGN DOCUMENT

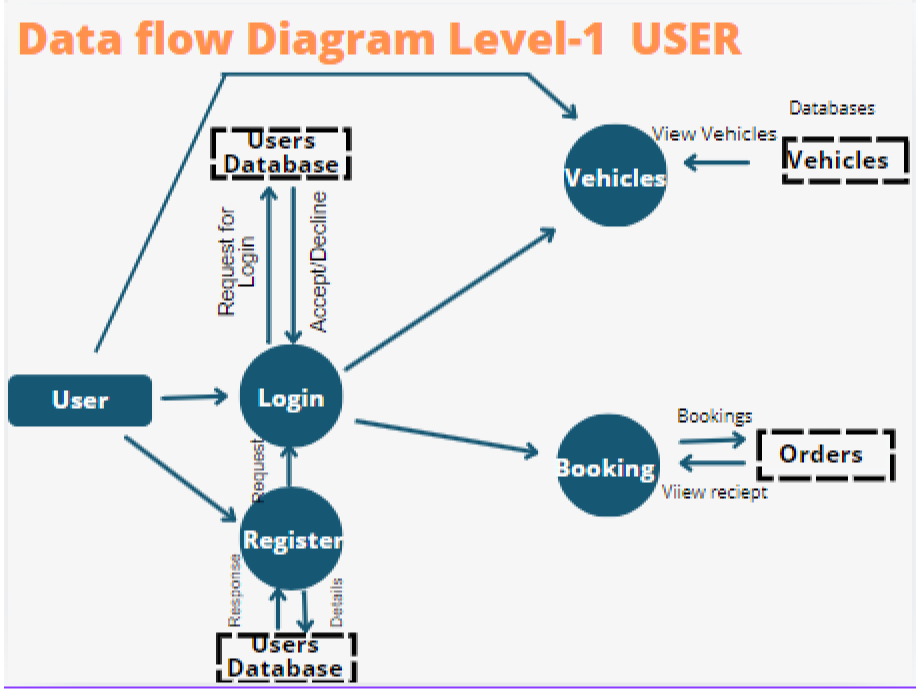
WORK FLOW DIAGRAM :



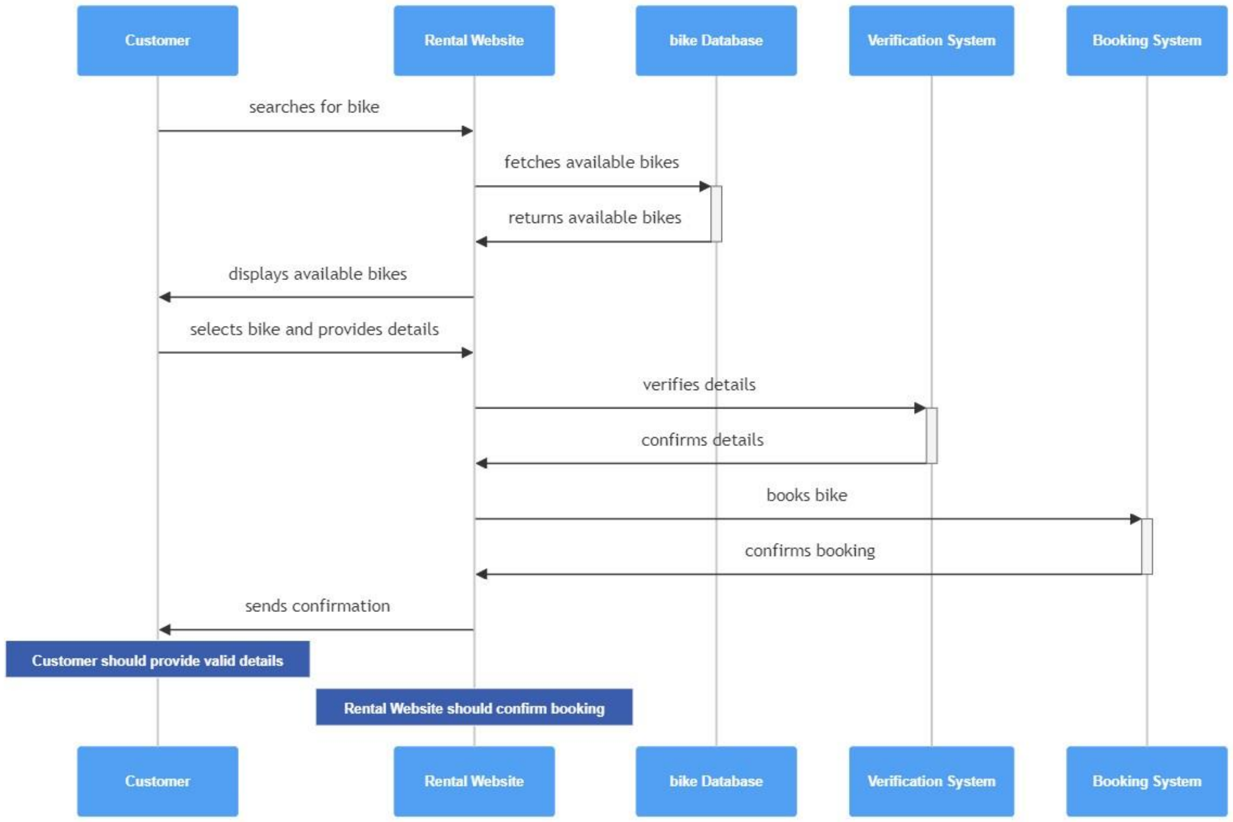
DATA FLOW DIAGRAM :



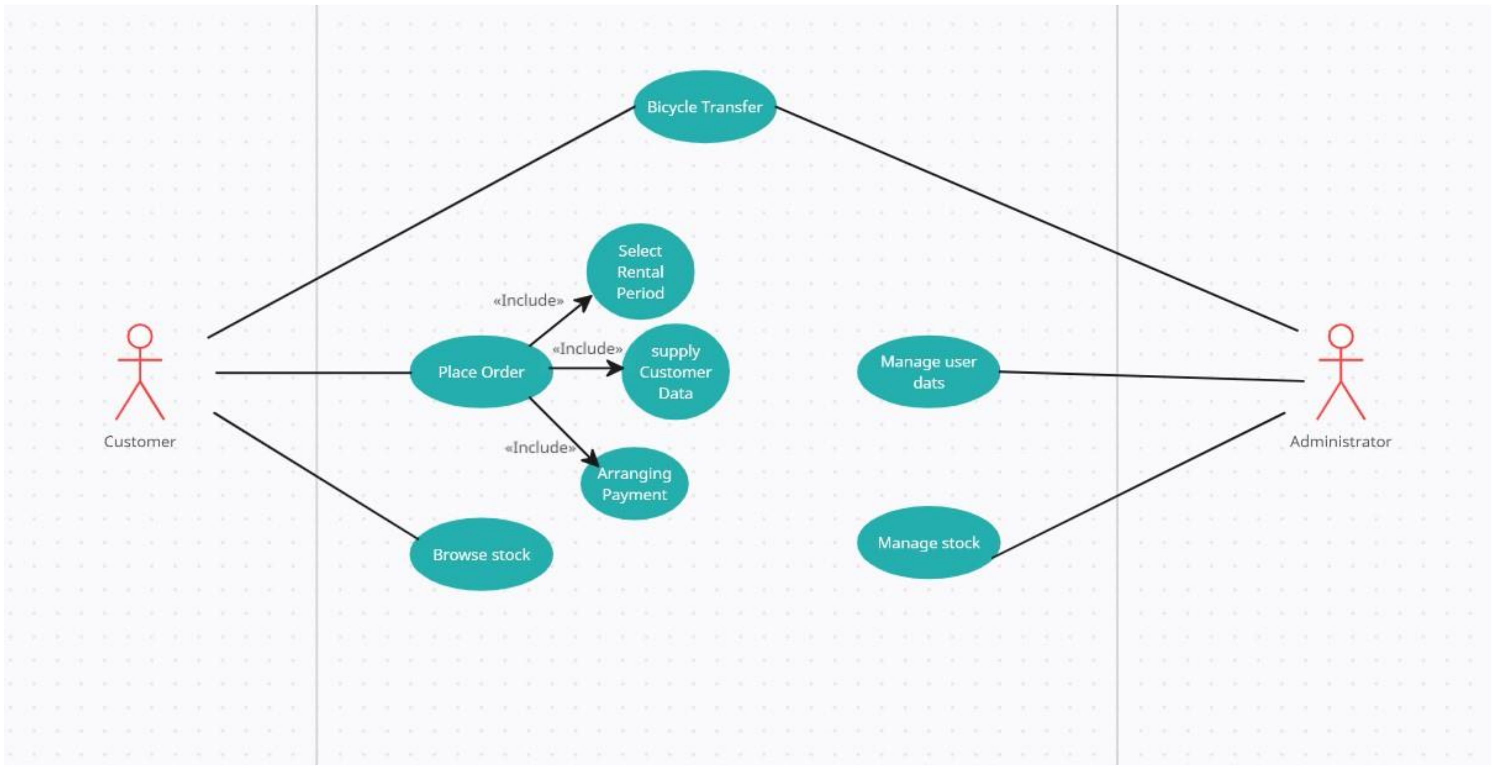




SEQUENCE DIAGRAM :



USE CASE DIAGRAM:


Bicycle Transfer
Select Rental Period
«Include»
«Include»
supply Customer Data
Place Order
«Include»
Arranging Payment
Browse stock
Manage user dats
Manage stock
Customer
Administrator