OCCUPATIONAL SPECIALISM

Ecocycle Rentals

Abstract

This document provides a useful guide for the design and development of a digital solution for EcoCycle Rentals, a company specialising in electric bicycle rentals. The design covers two key activities: Activity A (24 Marks), which focuses on problem identification, solution proposal, and requirements gathering, and Activity B (34 Marks), which details the creation of wireframes, algorithms, data models, and a test strategy for the system.

In Activity A, you will break down the main problems facing EcoCycle Rentals, propose solutions, justify design decisions, and outline functional and non-functional requirements, ensuring the system complies with relevant legal and regulatory standards. In Activity B, you will develop wireframes for the user interface, design algorithms to solve key problems, define data requirements, and create a comprehensive testing strategy.

The document aims to guide both technical and non-technical stakeholders through the solution's design, ensuring clarity and professionalism throughout.

Contents

Task One: Proposal

Scenario

You are working for a software development company, and your task is to design a digital solution for **EcoCycle Rentals**, a company that rents electric bicycles. EcoCycle Rentals wants a new system that will allow customers to:

- Book electric bikes online
- Track their bike using GPS while riding
- Earn loyalty points for every rental

Your design needs to cover the **booking system**, **GPS tracking**, and a **loyalty rewards program**. The system must be easy to use, secure, and allow customers to track their rentals in real-time. You are responsible for creating wireframes, algorithms, and a test strategy, and explaining how your design will work.

Important Notes for Students

- This template is designed to **help you structure your work** for the project.
- Follow each section carefully and use the **prompts** to guide your explanations.
- If you find writing challenging, try using **sentence starters** provided in each section.
- If you struggle with diagrams or explanations, feel free to ask for **additional support** or try breaking down the task into smaller pieces.

Activity A: Proposal Template (24 marks)

1.1. Identification of Problem (250 words)

Describe the main issue or challenge that EcoCycle Rentals is facing.

- **Prompt:** Clearly define the problem that your system will solve. Include background information, such as why the current system is insufficient, and explain why it is important to address this issue. *Sentence starter:* "The main problem that EcoCycle Rentals is experiencing is that their current system does not efficiently handle online bookings or provide real-time GPS tracking..."

1.2. Problem Decomposition (250 words)

Break the problem down into smaller, manageable parts.

- **Prompt:** Break the problem into sub-problems that need to be solved for the system to work efficiently. Explain each part and why it needs to be addressed. *Example:* "One issue is the lack of integration between bike availability and real-time tracking, which causes delays in updates for customers..."

1.3. Proposed Solution (400 words)

Present a detailed solution that will solve the identified problem.

- **Prompt:** Describe your proposed solution and how it addresses each part of the problem. Highlight key features and technologies that will be used. *Sentence starter:* "The proposed solution includes a fully integrated booking system that allows users to track their bike in real-time using GPS, alongside a loyalty rewards program..."

1.4. Justification of Choices (250 words)

Explain why you chose this solution and why it is the best fit.

- **Prompt:** Provide reasoning for your chosen solution and mention any alternatives you considered. Explain why your choice is the most effective based on feasibility, cost, or other factors. *Example:* "I chose this solution because it provides real-time data integration at a low cost while ensuring the system is scalable..."

1.5. User Function Table (10 functions)

Create a table of the main user functions.

- **Prompt:** List the key functions of the system, what different users (e.g., customers, admins) will do, and how the system will fulfil each function. Aim to include at least two types of users. *Example:*

Function	User Role(s)	Description of Functionality
Booking	Customer	Allows customers to book bikes online
GPS Tracking	Customer, Admin	Provides real-time location tracking of rented bicycles

1.6. Risk Table (5 risks)

Identify risks associated with the proposed solution and suggest mitigation strategies.

- **Prompt:** Use the table below to describe risks, their likelihood, impact, and how they will be mitigated. *Example:*

Risk	Likelihood	Impact	Mitigation Strategy
Data Breach	Medium	High	Use encryption for all customer data
System Downtime	Low	Medium	Implement regular system maintenance and updates

1.7. Justification of Risk Decisions (250 words)

Explain the reasoning behind the risk decisions and mitigation strategies.

- **Prompt:** For each risk, explain why you chose the mitigation strategy and how it will effectively minimise the impact of the risk. *Sentence starter:* "I rated the risk of a data breach as high due to the sensitivity of customer information. Encryption is essential to..."

1.8. Functional Requirements Table

Detail the functional requirements of the system in table format.

- **Prompt:** List the key functions that the system must perform. Each function should include a description and any constraints that might limit it. *Example:*

Requirement	**Description**	**Constraints**
User Login	Users must be able to log in with secure credentials	Must be encrypted
Real-Time GPS	Track the location of bikes	Must update location every 5 seconds

1.9. Non-Functional Requirements Table

List the non-functional aspects of the system such as performance, reliability, and security.

- **Prompt:** Use the table below to outline the non-functional requirements, such as performance standards and security expectations. *Example:*

Non-Functional Requirement	**Description**	**Expected Standard**
Performance	The system must be able to handle multiple users	Response time under 2 seconds
Security	Data encryption must be used for all transactions	SSL/TLS encryption
Usability	The system should be user-friendly	Error rate below 3% for users

1.10. Regulatory and Legal Requirements (250 words)

List any regulations or legal standards the system must comply with.

- **Prompt:** Detail the UK regulatory and legal standards that the system must follow (e.g., GDPR, accessibility, data protection). Explain how your system will comply with these regulations. *Example:* "To comply with GDPR, the system will ensure all personal data is encrypted and securely stored, and only accessible by authorised users."
- **Regulatory Standards to Consider:**
    - Data Protection Act 2018 (GDPR)
    - Equality Act 2010
    - Computer Misuse Act 1990
    - The Freedom of Information Act 2000
    - The Privacy and Electronic Communications Regulations (PECR)
    - Health and Safety at Work Act 1974

- The Consumer Rights Act 2015
- Accessibility Regulations 2018 (Public Sector Bodies Accessibility Regulations)
- Copyright, Designs and Patents Act 1988
- The Electronic Communications Act 2000

1.11. Key Performance Indicators (KPIs) (5 KPIs)

- Define metrics that will be used to measure the success of the project. These indicators should align with the project's objectives and be quantifiable.

KPI	**Description**
KPI 1	Description of the KPI and how it is measured
KPI 2	Description of another KPI

1.12. User Acceptance Criteria (10 items)

- List the criteria that must be met for the solution to be accepted by end-users. These are often measurable and used in testing to ensure that user needs are satisfied.

Acceptance Criteria	**Description**
Criteria 1	The system must allow users to register and receive a confirmation email within 5 minutes of signing up
Criteria 2	Description of another acceptance criterion

1.13. Site Map/Hierarchy Diagram

- Include a visual representation showing the structure of the system or website. This diagram should illustrate the relationships between key sections or pages, helping users understand the overall organisation. Use a tool like ‘Miro’ to create this diagram.

Activity B: Design Task Template (34 Marks)

2. Visual Interface (6 Marks)

Design a visual interface that includes wireframes for the home page, login page, and three functional pages. You must also explain your layout choices and provide a colour palette.

2.1 Wireframe of Home Page

Design a wireframe for the home page of your system.

- **Prompt:** Create a simple layout of your home page, showing the navigation bar, main buttons, and any other important sections.
  Example: "The home page allows users to quickly access key functions such as bike booking and GPS tracking."
- **Explanation of Home Page**
  Describe what each section of the home page does and why you placed elements in those locations.
  Example: "I placed the navigation bar at the top to ensure easy access to the most frequently used features."

2.2 Wireframe of Login Page

Create a wireframe for the login page.

- **Prompt:** Design the layout for users to enter their username, password, and login securely.
- **Explanation of Login Page**
  Explain how the login page ensures security and usability.
  Example: "The login page uses encrypted fields to securely capture user credentials, and a 'forgot password' link is provided for user convenience."

2.3 Wireframe of Functional Page 1 (FP1)

Create a wireframe for the first functional page, such as the booking page.

- **Prompt:** Design the page where users can book bikes and select rental options.
- **Explanation of FP1**
  Explain how users will interact with this page.
  Example: "The booking page allows users to select the bike they want to rent, the rental duration, and view real-time availability."

2.4 Wireframe of Functional Page 2 (FP2)

Create a wireframe for the second functional page, such as GPS tracking.

- **Prompt:** Design the page where users can track their bike’s location in real time.
- **Explanation of FP2**
  Explain why this feature is important to the user.
  Example: "The GPS tracking page displays the current location of the rented bike, allowing the user to monitor their journey and identify popular routes."

2.5 Wireframe of Functional Page 3 (FP3)

Create a wireframe for the third functional page, such as the loyalty rewards program.

- **Prompt:** Design the page where users can view and redeem their loyalty points.
- **Explanation of FP3**
  Explain how users can interact with the loyalty rewards program.
  Example: "The loyalty rewards page displays points earned from rentals and provides options for redeeming them for discounts."

2.6 Explanation of Layout Choices (250 words)

Explain the choices you made for the layout and design of your wireframes.

- **Prompt:** Describe why you placed certain elements (e.g., buttons, fields) in specific locations. Think about ease of use, accessibility, and overall look. *Example:* "I designed the navigation to be consistent across all pages to ensure a familiar and intuitive user experience."

2.7 Colour Palette

Choose a colour palette for the user interface and explain your choices.

- **Prompt:** Describe how your colour choices reflect EcoCycle Rentals' brand identity and enhance user experience. *Example:* "I chose a green and blue palette to represent sustainability and eco-friendliness, aligning with EcoCycle Rentals' branding." Use 'Adobe Color' to select the palette.

3. Algorithm Design (10 Marks)

Design the algorithms that will solve the key problems of the system. Include flowcharts, pseudocode, and explanations.

3.1 Problem Decomposition (250 words)

Break the problem down into smaller, manageable parts.

- **Prompt:** Explain the key sub-problems the system needs to solve, such as user authentication and bike booking. *Example:* "One sub-problem is ensuring that only authorised users can access the system. This requires a secure login process."

3.2 Inputs, Process, and Outputs

Describe the system's inputs, processes, and outputs for key functions.

- **Prompt:** For each function, describe what information the user provides (inputs), what the system does with this information (process), and what result the user receives (output). Add more rows as required.

Function	**Input**	**Process**	**Output**
Login	Username and password	The system checks the credentials	User is granted access to their account or user is asked to check details and try again.

3.3 Flowchart for Login Process

Create a flowchart that shows the steps involved in the login process.

- **Prompt:** Show the decisions made by the system, such as verifying credentials and granting or denying access.

3.4 Pseudocode for Login Algorithm

Write the pseudocode for the login process.

- **Prompt:** Break down the login process into simple, logical steps the system will follow. *Example pseudocode covered in Week 06.*

3.5 Sub-Process 1: Booking System

Design the booking system algorithm.

- **Prompt:** Use a flowchart and pseudocode to show how the booking system works. Include steps like selecting a bike, confirming availability, and completing the booking. *Example pseudocode and flowchart covered in Week 06.*

4. Data Requirements (6 Marks)

Describe the data your system will use and how it will be stored.

4.1 Table/Entity Design

Describe entities that should be included in your design.

- **Prompt:** Create list of tables and explain purpose. *Example:*

Table Name	**Purpose**
Customers	
Bookings	
Loyalty Points	

4.2 Data Dictionary

Create a data dictionary that defines each field in your tables.

- **Prompt:** For each field, describe its purpose, data type, and any constraints (e.g., primary key, foreign key).
  Example:

Table	Field	Description	Data Type	Constraints / Validation
customers	customer_id	Unique identifier	Integer	Primary Key, EXISTS, AUTOINCREMENT
customers	customer_username	Unique username	Text	Cannot be null, unique
customers	customer_email	Unique email address	Text	Cannot be null, unique
customers	customer_mobile	Unique mobile number	Text	Cannot be null, unique
customers	customer_password	Secure hash of password	Text	Cannot be null
customers	customer_createddate	Date of account creation	Date	Cannot be null
customers	customer_archivedate	Date of account archive	Date	
bookings	booking_id	Unique identifier	Integer	Primary Key, EXISTS, AUTOINCREMENT
bookings	customer_id	Link to customers table	Integer	Foreign Key, cannot be null

4.3 Data Flow Diagram

Create a data flow diagram that shows how data moves through the system.

- **Prompt:** Draw a diagram that shows how data is entered, processed, and output for key functions like booking, GPS tracking, and loyalty points. Use simple shapes to represent inputs, processes, and outputs, and arrows to show the flow of data.
  Example:
    - User enters booking details → System checks bike availability → Confirmation sent to user → Data stored in the booking table

4.4 Entity Relationship Diagram

Create an entity relationship diagram that shows the relationships between each table. Use ‘Miro’ to create your design.

5. Test Strategy (6 Marks)

Explain how you will test the system to ensure it works as intended.

5.1 Explanation of Test Strategy (250 words)

Describe the overall testing strategy for the system.

- **Prompt:** Explain how you will test individual components and the system as a whole. This could include unit testing, integration testing, and user acceptance testing.
  Example: "We will use unit testing to verify individual components like the login and booking systems, and integration testing to ensure they work together correctly."

5.2 Interrelation of Components (250 words)

Explain how different parts of the system interact and how this will be tested.

- **Prompt:** Describe how you will test the connections between various components (e.g., the connection between the booking system and GPS tracking).
  Example: "We will test that after a successful booking, the GPS tracking page correctly updates with the bike's location."

5.3 Transition from Unit to Integration Testing (100 words)

Describe the transition from unit testing to integration testing.

- **Prompt:** Explain how you will move from testing individual units to testing the system as a whole.
  Example: "Once individual units (e.g., login and booking) are tested, we will perform integration testing to ensure they work together without errors."

5.4 Functional and Non-Functional Testing (250 words)

Explain the difference between functional and non-functional testing and how both will be conducted.

- **Prompt:** Describe how you will test the functionality of key system features, as well as the performance, security, and usability of the system.
  Example:
    - **Functional Tests:** Ensure that the booking process correctly completes and updates the database.
    - **Non-Functional Tests:** Test how the system performs under heavy load and ensure it meets security standards, such as encryption.

5.5 Front-End and Back-End Tests (250 words)

Describe how you will test both the front-end and back-end components of the system.

- **Prompt:** Explain how you will test the user interface (front-end) and how the back-end processes data.
  Example: "We will test that the user interface is easy to navigate and that data entered on the front-end is correctly processed and stored by the back-end."

5.6 Order of Tests

List the order in which tests will be performed.

- **Prompt:** Outline the sequence of tests, starting with unit tests, followed by integration, functional, and non-functional tests.
  Example:
    1. Unit testing of individual components
    2. Integration testing of interconnected components
    3. Functional testing to ensure key features work
    4. Non-functional testing for performance and security

5.7 Acceptance, Alpha, and Beta Testing (250 words)

Describe how you will conduct acceptance, alpha, and beta testing.

- **Prompt:** Explain how you will perform these stages of testing, which involve user feedback and final adjustments.
  Example:
  - **Alpha Testing:** Initial internal testing to catch major issues before user testing.
  - **Beta Testing:** A limited group of users will test the system in real-world conditions.
  - **Acceptance Testing:** The final stage where the system is tested against user requirements to ensure it meets expectations.

5.8 Black Box and White Box Testing (250 words)

Explain how you will use both black box and white box testing.

- **Prompt:** Describe the difference between these two testing methods and how you will apply them.
  Example:
  - **Black Box Testing:** We will test the system's functionality without looking at the code, focusing on the user experience.
  - **White Box Testing:** We will test the internal logic and code structure to ensure it is efficient and error-free.

6. Design Document (6 Marks)

Use this section as a checklist to ensure you have met this criteria in the first part of this document.

Criteria	Prompt	Check
Document Suitability	Ensure your design document is clear and suitable for your intended audience.	
Clarity and Structure	Use headings, bullet points, and diagrams to organise your document. Ensure logical flow.	
Use of Technical Language	Use clear and concise technical terms, explaining complex concepts for non-technical stakeholders.	
Presentation Style and Consistency	Maintain a consistent visual style throughout your document. Use the same fonts, colours, and headings.	
Audience Suitability	Tailor the document to the needs of both technical and non-technical audiences.	
Final Checklist for Design Task	Is your document clearly structured with consistent headings and presentation style?	
	Have you used appropriate tools (e.g. Figma/Miro) to create professional wireframes and diagrams?	
	Is the content accessible to both technical and non-technical readers?	
	Have you completed wireframes and explained each page?	
	Did you include diagrams and explanations for all processes?	
	Is your test strategy complete and clear?	
	Is your design document clear for both technical and non-technical readers?	