HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

School of Information and communications technology

Software Design Document

Tài liệu thiết kế

THIẾT KẾ VÀ XÂY DỰNG APP ECO\_BIKE\_RENTAL

Môn: Thiết kế và xây dựng phần mềm

Nhóm 21

Lê Phương Nam: 20183957

Lê Minh Nghĩa: 20183960

Bùi Tiến Đạt: 20183880

Eang Sokunthea: 20180280

*Hà Nội, ngày 4 tháng 1 năm 2022*

Contents

[Table of Contents 1](#_Toc92161776)

[1. Introduction 5](#_Toc92161777)

[1.1. Objective 5](#_Toc92161778)

[1.2. Scope 5](#_Toc92161779)

[1.3. References 5](#_Toc92161780)

[2. Overall Description 6](#_Toc92161781)

[2.1. General Overview 6](#_Toc92161782)

[2.2. Assumptions/Constraints/Risks 6](#_Toc92161783)

[2.2.1. Assumptions 6](#_Toc92161784)

[2.2.2. Constraints 6](#_Toc92161785)

[2.2.3. Risks 7](#_Toc92161786)

[3. System Architecture and Architecture Design 8](#_Toc92161787)

[3.1. Architectural Patterns 8](#_Toc92161788)

[3.2. Interaction Diagrams 8](#_Toc92161789)

[3.3. Analysis Class Diagrams 8](#_Toc92161790)

[3.4. Unified Analysis Class Diagram 8](#_Toc92161791)

[3.5. Security Software Architecture 8](#_Toc92161792)

[4. Detailed Design 9](#_Toc92161793)

[4.1. User Interface Design 9](#_Toc92161794)

[4.1.1. Screen Configuration Standardization 9](#_Toc92161795)

[4.1.2. Screen Transition Diagrams 9](#_Toc92161796)

[4.1.3. Screen Specifications 9](#_Toc92161797)

[4.2. Data Modeling 9](#_Toc92161798)

[4.2.1. Conceptual Data Modeling 9](#_Toc92161799)

[4.2.2. Database Design 9](#_Toc92161800)

[4.3. Non-Database Management System Files 10](#_Toc92161801)

[4.4. Class Design 10](#_Toc92161802)

[4.4.1. General Class Diagram 10](#_Toc92161803)

[4.4.2. Class Diagrams 10](#_Toc92161804)

[4.4.3. Class Design 11](#_Toc92161805)

[5. Design Considerations 13](#_Toc92161806)

[5.1. Goals and Guidelines 13](#_Toc92161807)

[5.2. Architectural Strategies 13](#_Toc92161808)

[5.3. Coupling and Cohesion 14](#_Toc92161809)

[5.4. Design Principles 14](#_Toc92161810)

[5.5. Design Patterns 14](#_Toc92161811)

**List of Figures**

**No table of figures entries found.**

**List of Tables**

No table of figures entries found.

# Introduction

*<The following subsections of the Software Design Document (SDD) document should provide an overview of the entire SDD.>*

## Objective

Tài liệu đặc tả chi tiết các use case của ứng dụng EcoBikeRetal bao gồm: xem thông tin bãi xe, xem thông tin chi tiết của xe, thuê xe và trả xe. Tài liệu mô tả mục đích và các tính năng của hệ thống, các giao diện, ràng buộc của hệ thống cần được thực hiện để phản ứng với các kích thích bên ngoài.

## Scope

Khu đô thị Ecopark có dịch vụ cho thuê xe đạp theo giờ với nhiều bãi để thuê xe và trả xe tự động trong khu đô thi.

Khi vào ứng dụng, ứng dụng sẽ hiển thị danh sách các bãi đỗ xe gần với vị trí của khách hàng nhất, lúc đó khách hàng sẽ chọn bất kỳ một bãi đỗ xe hiển thị trong danh sách và xem thông tin về bãi xe. Đống thời khách hàng cũng có thể xem chi tiết từng xe trong bãi đậu xe.

Để có thể thuê xe, khách hàng cần sử dụng ứng dụng EcoBikeRental để nhập mã khóa xe. Sau khi xác nhận mã, hệ thống sẽ hiển thị thông tin về xe và lúc này khách hàng sẽ thực hiện đặt cọc tiền để có thể thuê xe.

Trong quá trình thuê xe khách hàng có thể xem thông tin chi tiết về xe. Khi cần trả xe, khách hàng đưa xe vào vị trí trống bất kì trong bãi bất kì và đóng khóa xe lại. Lúc này hệ thống sẽ tự động trả lại tiền đặt cọc xe và trừ đi số tiền mà khách đã thuê.

## References

|  |  |
| --- | --- |
| [1] | Centers for Medicare & Medicaid Services, "System Design Document Template," [Online]. Available: https://www.cms.gov/Research-Statistics-Data-and-Systems/CMS-Information-Technology/XLC/Downloads/SystemDesignDocument.docx. |

# Overall Description

<*This section describes the principles and strategies to be used as guidelines when designing and implementing the system.>*

## General Overview

*<Briefly introduce the system context and the basic design approach or organization. Provide a brief overview of the system and software architectures and the design goals. Include the high-level context diagram(s) for the system and subsystems provided in previous documents like SRS (e.g., general use case diagram, lower-level use case diagrams, activity diagrams), updated as necessary to reflect any changes that have been made based on more current information or understanding. If the high-level context diagram has been updated, identify the changes that were made and why>*

## Assumptions/Constraints/Risks

### Assumptions

*<Describe any assumptions or dependencies regarding the system, software and its use. These may concern such issues as: related software or hardware, operating systems, end-user characteristics, and possible and/or probable changes in functionality>*

### Constraints

*<Describe any global limitations or constraints that have a significant impact on the design of the system’s hardware, software and/or communications, and describe the associated impact. Such constraints may be imposed by any of the following (the list is not exhaustive):*

* *Hardware or software environment*
* *End-user environment*
* *Availability or volatility of resources*
* *Standards compliance*
* *Interoperability requirements*
* *Interface/protocol requirements*
* *Licensing requirements*
* *Data repository and distribution requirements*
* *Security requirements (or other such regulations)*
* *Memory or other capacity limitations*
* *Performance requirements*
* *Network communications*
* *Verification and validation requirements (testing)*
* *Other means of addressing quality goals*
* *Other requirements described in the Requirements Document*

*>*

### Risks

*<Describe any risks associated with the system design and proposed mitigation strategies.>*

# System Architecture and Architecture Design

<*Briefly describe the architectural design steps*>

## Architectural Patterns

*<Specify and briefly describe the chosen architectural patterns and the reasons why they were chosen>*

## Interaction Diagrams

## Analysis Class Diagrams

## Unified Analysis Class Diagram (gop)

## Security Software Architecture

*<Describe the software components and configuration supporting the security and privacy of the system. Specify the architecture for (1) authentication to validate user identity before allowing access to the system;(2) authorization of users to perform functional activity once logged into the system, (3) encryption protocol to support the business risks and the nature of information, and (4) logging and auditing design, if required.>*

# Detailed Design

## User Interface Design

*<Suppose that you design a Graphical User Interface (GUI)>*

### Screen Configuration Standardization

### Screen Transition Diagrams

### Screen Specifications

*<Screen images should be included in the screen specifications>*

## Data Modeling

### Conceptual Data Modeling

*<E-R Diagram image and description of entities and relationships>*

Diagram, schematic

Description automatically generated

### Database Design

#### Database Management System

*<Specify what is the decision of Database Management System (DBMS) and give some description of the DBMS>*

#### Database Diagram

<

* *Show the process to design database from E-R diagram*
* *Show the diagram of DB design*

*>*

Diagram

Description automatically generated

#### Database Detail Design

1 Station

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **STT** | **PK** | **FK** | **Tên cột** | **Loại dữ liệu** | **Bắt buộc** | **Mô tả** |
| 1 | x |  | id | INT | Có | Id, not null, auto increment |
| 2 |  |  | name | VARCHAR | Có | Tên bãi xe |
| 3 |  |  | address | VARCHAR | Có | Địa chỉ bãi xe |
| 4 |  |  | area | VARCHAR | Có | Diện tích bãi xe |
| 5 |  |  | num\_of\_curr\_bike | INT | Có | Số lượng xe hiện có trong bãi |
| 6 |  |  | num\_of\_blank\_bike | INT | Có | Số lượng ô trống của xe đạp |
| 7 |  |  | num\_of\_blank\_electric\_bike | INT | Có | Số lượng ô trống của xe đạp |
| 8 |  |  | num\_of\_blank\_twin\_bike | INT | Có | Số lượng ô trống xủa xe đạp đôi thường |
| 9 |  |  | num\_of\_blank\_electric\_twin\_bike | INT | Có | Số lượng ô trống xủa xe đạp đôi điện |

2 ParkingSlot

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **STT** | **PK** | **FK** | **Tên cột** | **Loại dữ liệu** | **Bắt buộc** | **Mô tả** |
| 1 | x |  | id | INT | Có | Id, not null, auto increment |
| 2 |  |  | type | INT | Có | Loại ô trống |
| 3 |  |  | code | VARCHAR | Có | Mã khóa xe ở ô trống |
| 4 |  |  | status | INT | Có | Trạng thái của bãi xe (Còn trống, Có xe, Có xe đang đặt trả) |
| 5 |  |  | bookingTime | BIGINT | Không | Thời gian đặt xe để trả |
| 6 |  | x | stationId | INT | Có | Id của bãi xe |

3 Vehicle

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **STT** | **PK** | **FK** | **Tên cột** | **Loại dữ liệu** | **Bắt buộc** | **Mô tả** |
| 1 | x |  | id | INT | Có | Id, not null, auto increment |
| 2 |  |  | type | INT | Có | Loại phương tiện |
| 3 |  |  | liense\_plate | VARCHAR | Có | Biển số xe |
| 4 |  |  | status | INT | Có | Trạng thái của xe |
| 5 |  |  | battery | INT | Không | Phần trăm pin của xe điện |
| 6 |  |  | max\_time | INT | Không | Thời gian tối đa của xe điện có thể sử dụng |
| 7 |  |  | parkingslot\_id | INT | Không | Vị trí ô trống xe đạp đang đỗ |

4 Card

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **STT** | **PK** | **FK** | **Tên cột** | **Loại dữ liệu** | **Bắt buộc** | **Mô tả** |
| 1 | x |  | id | INT | Có | Id, not null, auto increment |
| 2 |  |  | card\_code | VARCHAR | Có | Mã số thẻ |
| 3 |  |  | owner | VARCHAR | Có | Tên chủ sở hữu |
| 4 |  |  | date\_expired | BIGINT | Có | Ngày hết hạn |
| 5 |  |  | cvv\_code | VARCHAR | Có | Mã bảo mật |
| 7 |  |  | status | INT | Có | Trạng thái của thẻ |

5 PaymentTransaction

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **STT** | **PK** | **FK** | **Tên cột** | **Loại dữ liệu** | **Bắt buộc** | **Mô tả** |
| 1 | x |  | id | INT | Có | Id, not null, auto increment |
| 2 |  |  | method | VARCHAR | Có | Phương thức thanh toán |
| 3 |  |  | content | VARCHAR | Có | Nội dung thanh toán |
| 4 |  |  | createAt | BIGINT | Có | Thời điểm thanh toán |
| 5 |  | x | card\_id | INT | Có | Thẻ thanh toán |

6 Invoice

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **STT** | **PK** | **FK** | **Tên cột** | **Loại dữ liệu** | **Bắt buộc** | **Mô tả** |
| 1 | x |  | id | INT | Có | Id, not null, auto increment |
| 2 |  |  | start\_time | BIGINT | Có | Thời gian bắt đầu thuê xe |
| 3 |  |  | restart\_time | BIGINT | Không | Thời gian bắt đầu thuê lại xe gần nhất |
| 4 |  |  | total\_rent\_time | INT | Có | Tổng thời gian thuê xe |
| 5 |  |  | status | INT | Có | Trạng thái của hóa đơn |
| 6 |  |  | vehicle\_id | INT | Có | Phương tiện thuê |
| 7 |  | x | transaction\_id | INT | Có | Giao dịch |

**Database script**

CREATE DATABASE ECOBIKE;

create table stations

(

id int auto\_increment not null,

name varchar(100) CHARACTER SET utf8mb4 COLLATE utf8mb4\_bin DEFAULT NULL,

address varchar(250) CHARACTER SET utf8mb4 COLLATE utf8mb4\_bin DEFAULT NULL,

area varchar(250) CHARACTER SET utf8mb4 COLLATE utf8mb4\_bin DEFAULT NULL,

num\_of\_curr\_bike int DEFAULT NULL,

num\_of\_blank\_bike int DEFAULT NULL,

num\_of\_blank\_electric\_bike int DEFAULT NULL,

num\_of\_blank\_twin\_bike int DEFAULT NULL,

num\_of\_blank\_electric\_twin\_bike int DEFAULT NULL,

PRIMARY KEY (id)

);

create table parking\_slots

(

id int auto\_increment not null,

station\_id int DEFAULT NULL,

type int DEFAULT NULL,

code varchar(50) CHARACTER SET utf8mb4 COLLATE utf8mb4\_bin DEFAULT NULL,

status int DEFAULT NULL,

booking\_time bigint DEFAULT NULL,

PRIMARY KEY (id)

);

create table vehicles

(

id int auto\_increment not null,

parking\_slot\_id int DEFAULT NULL,

type int DEFAULT NULL,

license\_plate varchar(50) COLLATE utf8mb4\_bin DEFAULT NULL,

battery int DEFAULT NULL,

max\_time int DEFAULT NULL,

status int DEFAULT NULL,

PRIMARY KEY (id)

);

create table cards

(

id int auto\_increment not null,

card\_code varchar(50) CHARACTER SET utf8mb4 COLLATE utf8mb4\_bin DEFAULT NULL,

owner varchar(100) CHARACTER SET utf8mb4 COLLATE utf8mb4\_bin DEFAULT NULL,

date\_expired bigint DEFAULT NULL,

cvv\_code varchar(50) CHARACTER SET utf8mb4 COLLATE utf8mb4\_bin DEFAULT NULL,

status int DEFAULT NULL,

PRIMARY KEY (id)

);

create table payment\_transactions

(

id int auto\_increment not null,

card\_id int DEFAULT NULL,

method varchar(100) CHARACTER SET utf8mb4 COLLATE utf8mb4\_bin DEFAULT NULL,

content varchar(100) CHARACTER SET utf8mb4 COLLATE utf8mb4\_bin DEFAULT NULL,

created\_at bigint DEFAULT NULL,

PRIMARY KEY (id)

);

create table invoices

(

id int auto\_increment not null,

vehicle\_id int DEFAULT NULL,

transaction\_id int DEFAULT NULL,

start\_time bigint DEFAULT NULL,

restart\_time bigint DEFAULT NULL,

total\_rent\_time int DEFAULT NULL,

status int DEFAULT NULL,

PRIMARY KEY (id)

)

ALTER TABLE parking\_slots ADD FOREIGN KEY (station\_id) REFERENCES stations(id)

ALTER TABLE vehicles ADD FOREIGN KEY (parking\_slot\_id) REFERENCES parking\_slots(id)

ALTER TABLE invoices ADD FOREIGN KEY (vehicle\_id) REFERENCES vehicles(id)

ALTER TABLE invoices ADD FOREIGN KEY (transaction\_id) REFERENCES payment\_transactions(id)

ALTER TABLE payment\_transactions ADD FOREIGN KEY (card\_id) REFERENCES cards(id)

## Non-Database Management System Files

*<Provide the detailed description of all non-DBMS files if any and include a narrative description of the usage of each file that identifies if the file is used for input, output, or both, and if the file is a temporary file. Also provide an indication of which modules read and write the file and include file structures (refer to the data dictionary). As appropriate, the file structure information should include the following:*

*• Record structures, record keys or indexes, and data elements referenced within the records*

*• Record length (fixed or maximum variable length) and blocking factors*

*• Access method (e.g., index sequential, virtual sequential, random access, etc.)*

*• Estimate of the file size or volume of data within the file, including overhead resulting from file access methods*

*• Definition of the update frequency of the file (If the file is part of an online transaction-based system, provide the estimated number of transactions per unit of time, and the statistical mean, mode, and distribution of those transactions.)*

*• Backup and recovery specifications>*

## Class Design

### General Class Diagram

<General class diagram which shows the whole class diagram of the software. This diagram may have packages, subsystems and classes. Classes in this diagram may not have all attributes and operations>

### Class Diagrams

<Detail class diagram with full attributes and operations>

#### Class Diagram for Package A

#### Class Diagram for Subsystem B

…

### Class Design

<Detail design for each class>

#### Class “SampleClass1”

<SampleClass1 class image in UML>

Table 1. Example of attribute design

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *#* | *Name* | *Data type* | *Default value* | *Description* |
| 1 |  |  |  |  |
| 2 |  |  |  |  |

Table 1. Example of operation design

|  |  |  |  |
| --- | --- | --- | --- |
| *#* | *Name* | *Return type* | *Description (purpose)* |
| 1 |  |  |  |
| 2 |  |  |  |

*Parameter*:

* x: Default value, description
* y: Default value, description

*Exception*:

* AException if …
* BException if …

**Method**

How to use parameters / attributes

Flowchart / Sequence diagram if the method has a complex/special algorithm

**State**

State diagram if any

#### Class “SampleClass2”

…

# Design Considerations

***<Describe issues which need to be addressed or resolved before attempting to devise a complete design solution. Remember that, you have to refactor your source code to strictly follow the final design>***

## Goals and Guidelines

*<Describe any goals, guidelines, principles, or priorities which dominate or embody the design of the system and its software.*

*Examples of such goals might be: an emphasis on speed versus memory use; or working, looking, or “feeling” like an existing product.*

*Guidelines include coding guidelines and conventions.*

*For each such goal or guideline, describe the reason for its desirability unless it is implicitly obvious.*

*Describe any design policies and/or tactics that do not have sweeping architectural implications (meaning they would not significantly affect the overall organization of the system and its high-level structures), but which nonetheless affect the details of the interface and/or implementation of various aspects of the system (e.g., choice of which specific product to use)*>

## Architectural Strategies

*<Describe any design decisions and/or strategies that affect the overall organization of the system and its higher-level structures. These strategies should provide insight into the key abstractions and mechanisms used in the system architecture. Describe the reasoning employed for each decision and/or strategy (possibly referring to previously stated design goals and principles) and how any design goals or priorities were balanced or traded-off.*

*Examples of design decisions might concern (but are not limited to) things like the following:*

*• Use of a particular type of product (programming language, database, library, commercial off-the-shelf (COTS) product, etc.)*

*• Reuse of existing software components to implement various parts/features of the system*

*• Future plans for extending or enhancing the software*

*• User interface paradigms (or system input and output models)*

*• Hardware and/or software interface paradigms*

*• Error detection and recovery*

*• Memory management policies*

*• External databases and/or data storage management and persistence*

*• Distributed data or control over a network*

*• Generalized approaches to control*

*• Concurrency and synchronization*

*• Communication mechanisms*

*• Management of other resources*

>

## Coupling and Cohesion

*<Evaluate your design and describe which levels of coupling and cohesion that your design is at. Give proofs for your assumptions. Explain if there is any special design or exceptions>*

*<You may show the previous design from which you made improvements to get better levels of coupling and cohesion. You should clarify how and why you did these improvements>*

## Design Principles

*<Does your design follow the SOLID principles if there are new requirements/changing requirements in the future? Give proofs for your assumptions. Explain if there is any special design or exceptions>*

*<You may show the previous design from which you made improvements to get a better design, which follows SOLID principles in spite of additional requirements. You should clarify how and why you did these improvements>*

## Design Patterns

*<Do you use any design patterns for your design? If yes, describe detailly why you use those design patterns? Describe in detail on the solutions and how to implement each design pattern>*