HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

School of Information and communications technology

Software Design Document

Version 1.0

Capstone Project

Subject: EcoBikeRental

Group 19

|  |  |  |
| --- | --- | --- |
| Lê Minh Tuấn |  |  |
| Lý Bá Tuấn | 20173438 |  |
|  |  |  |
|  |  |  |

*Hanoi,* *12 / 2020*

*<All notations inside the angle bracket are not part of this document, for its purpose is for extra instruction. When using this document, please erase all these notations and/or replace them with corresponding content as instructed>*

*<This document, written by Asst. Prof. NGUYEN Thi Thu Trang, is used as a case study for student with related courses. Any modifications and/or utilization without the consent of the author is strictly forbidden>*

Table of Contents

Table of Contents 1

1 Introduction 3

1.1 Objective 3

1.2 Scope 3

1.3 Glossary 3

1.4 References 3

2 Overall Description 4

2.1 General Overview 4

2.2 Assumptions/Constraints/Risks 4

2.2.1 Assumptions 4

2.2.2 Constraints 4

2.2.3 Risks 5

3 System Architecture and Architecture Design 6

3.1 Architectural Patterns 6

3.2 Interaction Diagrams 6

3.3 Analysis Class Diagrams 6

3.4 Unified Analysis Class Diagram 6

3.5 Security Software Architecture 6

4 Detailed Design 7

4.1 User Interface Design 7

4.1.1 Screen Configuration Standardization 7

4.1.2 Screen Transition Diagrams 7

4.1.3 Screen Specifications 7

4.2 Data Modeling 7

4.2.1 Conceptual Data Modeling 7

4.2.2 Database Design 7

4.3 Non-Database Management System Files 8

4.4 Class Design 8

4.4.1 General Class Diagram 8

4.4.2 Class Diagrams 8

4.4.3 Class Design 8

5 Design Considerations 10

5.1 Goals and Guidelines 10

5.2 Architectural Strategies 10

5.3 Coupling and Cohesion 11

5.4 Design Principles 11

5.5 Design Patterns 11

**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

<*Identify the purpose of this SDD and its intended audience. In this subsection, describe the purpose of the SDD and specify the intended audience for the SDD*>

## Scope

<*In this subsection:*

1. *Identify the software product(s) to be produced by name*
2. *Explain what the software product(s) will, and, if necessary, will not do*
3. *Describe the application of the software being specified, including relevant benefits, objectives, and goals*
4. *Be consistent with similar statements in higher-level specifications if they exist*

*This should be an executive-level summary. Do not enumerate the whole requirements list here*

*Note that this will be similar to what was written in the SRS.*

>

## Glossary

*<Listing and explaining the terms appearing in the software’s profession and this document. Any assumption of the reader’s prior knowledge or experience on the subject is ill advised>*

## References

Centers for Medicare & Medicaid Services. (n.d.). *System Design Document Template.* Retrieved from Centers for Medicare & Medicaid Services: https://www.cms.gov/Research-Statistics-Data-and-Systems/CMS-Information-Technology/XLC/Downloads/SystemDesignDocument.docx

*<Listing the referenced material used in this document, including the one related to the project>*

# Overall Description

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

## General Overview

*Tài liệu này mô tả các thiết kế chi tiết hệ thống thuê xe tự động EcoBikeRental, phục vụ hoạt động trong khu đô thị EcoPark. Tài kiệu thiết kế theo chiều phát triển dần dần của ứng dụng, đi từ việc lựa chọn mẫu kiến trúc thiết kế sau đó tới xây dựng các biểu đồ trình tự biểu thị luồng hoạt động của các nghiệp vụ trong hệ thống một cách cụ thể và rõ ràng nhất. Sau đó, sẽ phân tích các lớp, rồi xây dựng thực thể, xây dựng database, giao diện và cuối cùng là thiết kế lớp chi tiết cùng với các pattern sử dụng trong source code viết nên hệ thống.*

## 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 và Architecture Design

<*Briefly describe the architectural design steps*>

Các bước thiết kế kiến trúc:

1. Xây dựng khung kiến trúc.
2. Từ tài liệu SRS, thiết kế luồng hoạt động tuần tự trong hệ thống qua biểu đồ trình tự (Sequence diagram).
3. Vẽ biểu đồ lớp phân tích sau khi hoàn thành biểu đồ trình tự.

## Mẫu kiến trúc sử dụng: Kiến trúc 3 tầng

* *Mô tả: Kiến trúc 3 tầng (three layer) chia ứng dụng ra làm 3 tầng chính là tầng trình diễn (presentation), tầng nghiệp vụ (business) và tầng dữ liệu (data). Tầng ứng dụng xử lý các nghiệp vụ logic. Tầng dữ liệu để lưu trữ thông tin, biểu diễn, trao đổi thông tin với Database. Tầng trình diễn xử lý và hiển thị giao diện lên cho người dùng.*
* *Lý do lựa chọn:*
  + *Kiến trúc 3 tầng dễ hiểu, không khó trong việc triển khai đồng thời dễ bảo trì khi các tầng được phân chia rõ ràng, tái sử dụng tốt.*
  + *Dễ dàng phân chia khối lượng công việc cho các thành viên trong nhóm theo các chức năng mà ít bị xung đột lẫn nhau.*

## Sequence Diagrams

## Analysis Class Diagrams

## Unified Analysis Class Diagram

# Detailed Design

## User Interface Design

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

### Screen Configuration Standardization

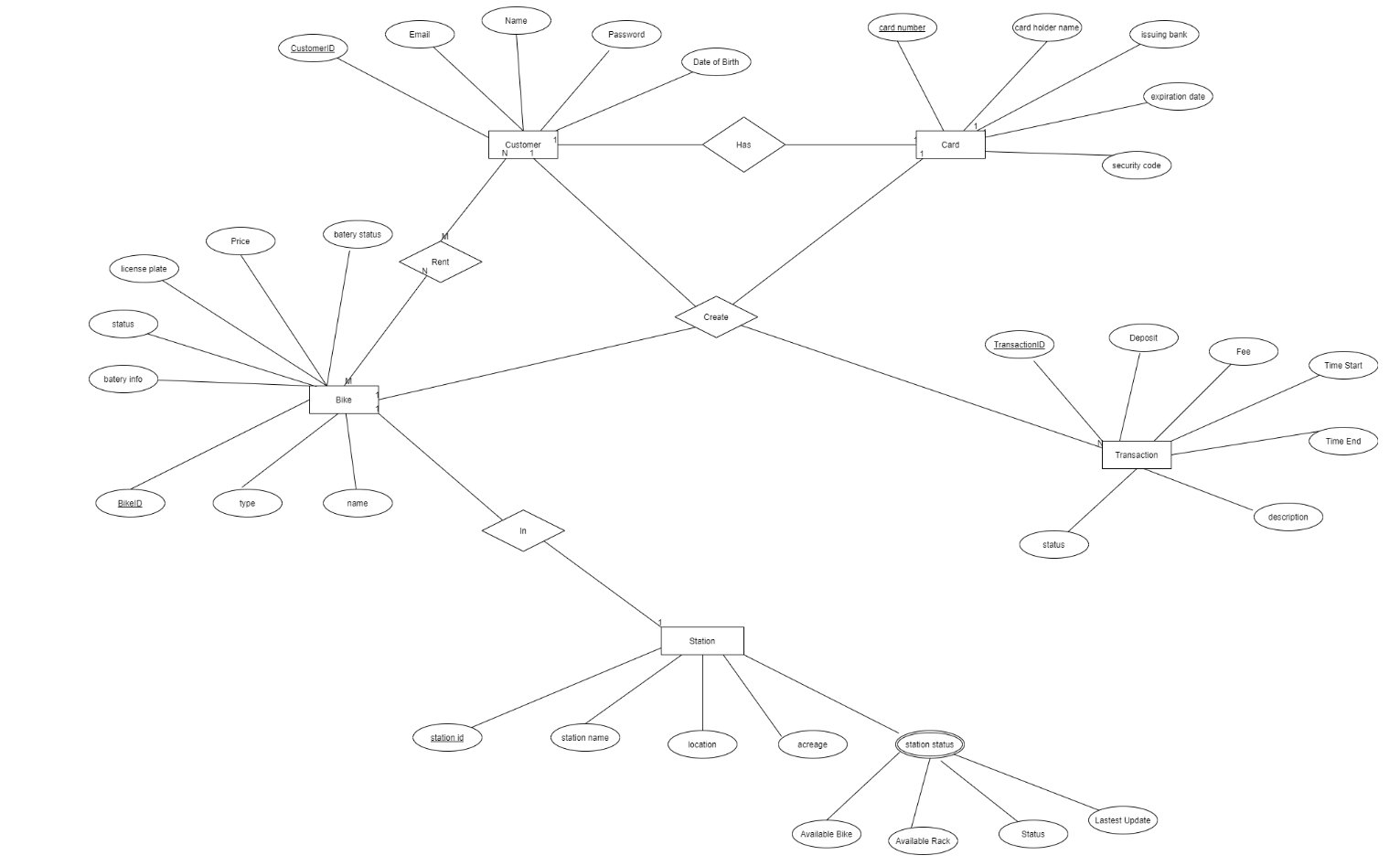
### Screen Transition Diagrams

### Screen Specifications

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

## Data Modeling

### Conceptual Data Modeling



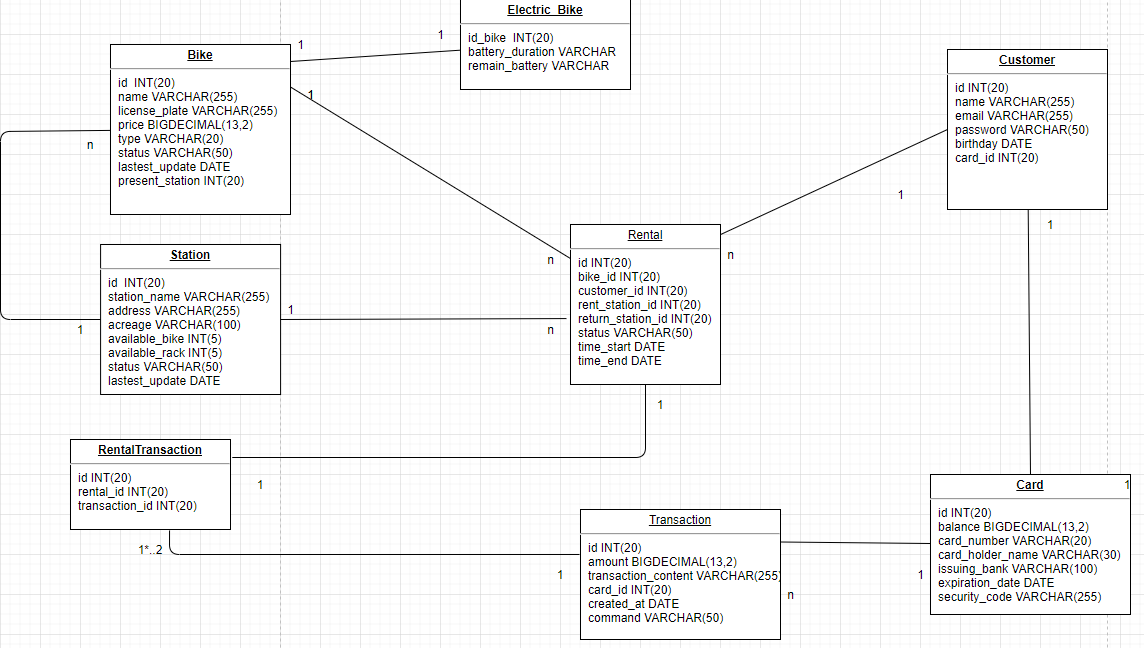
Hình … biểu diễn sơ đồ thực thể liên kết của database trong hệ thống. Bao gồm các quan hệ:

* Customer có Card
* Customer, Card, Bike kết hợp lại sẽ tạo thành một nhóm hoạt động thuê xe, trả xe và tạo Transaction.
* Bike sẽ nằm trong Station, 1 station có nhiều Bike.

### Database Design

#### Database Management Systems

#### Logical Data Model



Mô tả:

Bike: lưu trữ thông tin chung của các loại xe đạp. Khi xe đạp có một tính năng mới nổi bật thì có thể tạo thêm các bảng Bike mới như bảng Electric\_Bike có khóa ngoại tham chiếu đến bảng Bike. Đặc biệt ở Bike, có trường status biểu thị tình trạng xe là: đang thuê (RENTING\_BIKE) hay có thể thuê bất cứ lúc nào (AVAILABLE). Trường present\_station là khóa ngoại tham chiếu tới id của bảng Station thể hiện thông tin trạm xe hiện tại mà Bike đang được đặt.

Station: lưu trữ thông tin của trạm xe.

Customer: lưu trữ thông tin khách hàng

Card: lưu trữ thông tin credit card của khách hàng thuê xe

Transaction: lưu trữ các thông tin giao dịch được thực hiện.

Rental: bảng này chứa thông tin thuê xe của khách hàng bao gồm id của xe được thuê, id khách hàng, id của trạm mà được lấy xe ra, id của trạm mà xe được trả về, status hiện tại (đang thuê hay đã trả xe), thời gian bắt đầu tính thuê xe, thời gian kết trả xe.

RentalTransaction: bảng này là bảng quan hệ giữa hai bảng Rental và Transaction, biểu thị quan hệ một nhiều (một record của Rental sẽ có tối đa 2 record của Transaction do trong mỗi hoạt động thuê xe thì chỉ xảy ra 2 giao dịch là lúc đặt cọc xe và lúc thanh toán).

#### Physical Data Model

<

*Give a detail design of each element in the DB diagram. For instance, in a Relational DBMS, give a detail design for each Table and their constraints, illustrated in below table (PK: Primary Key, FK: Foreign Key).*

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| *#* | *PK* | *FK* | *Column name* | *Data type* | *Default value* | *Mandatory* | *Description* |
| 1 | x |  | ProductID |  |  |  |  |
| 2 |  | x | CategoryID |  |  |  |  |

*You may add indexing, trigger, view, etc.*

*Give the database script*>

## 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

### Class Diagrams

#### Class Diagram for Package A

#### Class Diagram for Subsystem B

…

### Class Design

#### Class “SampleClass1”

<SampleClass1 class image in UML>

**Attribute**

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

**Operation**

|  |  |  |  |
| --- | --- | --- | --- |
| *#* | *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>***

## 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>*

## Design Principles

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

## 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>*