**Canadian Institute of Technology**

Faculty of Engineering   
Software Engineering Department

**Bike Sharing System**

*A project submitted*

*in partial fulfillment of the requirements for the degree of*

*Bachelor Studies in* Software Engineering

**by**

Anisa Gurabardhi

**Supervised by**

Fabjan Lashi

**UNDERTAKING**

This is to declare that the project entitled “Bike Sharing System” is an original work done by the undersigned, Anisa Gurabardhi, in partial fulfillment of the requirements for the degree “Bachelor of Software Engineering"

All the analysis, design and system development have been accomplished by the undersigned, Anisa Gurabardhi. Moreover, this project has not been submitted to any other college or university.

Student: Anisa Gurabardhi

**ABSTRACT**

*This project deals with developing a desktop application for a Bike Sharing System. Currently speaking, we don't have a similar system in Albania for this bike renting activity. In order to facilitate the work of users and also employees I have decided to build a simple version. This system provides the users with a catalog of different bicycles (and also their locations) available for rent. It also provides the admin with options for managing all users and bicycles. The admin can remove, delete and also update any user or bicycle at any time. This system is implemented using Java as a programming language and a relation database such as MySQL. This system will solve the problem of registering and identifying each user more easily. Also users don't have to exchange their ID card for the amount of time they are using the bicycles.*

***Keywords****: Bike Sharing system, desktop application, Java, MySQL, user, admin*

Table of Contents

[LIST OF TABLES v](#_Toc507401060)

[LIST OF FIGURES vi](#_Toc507401061)

[1. Software Requirements 1](#_Toc507401062)

[*1.1 System purpose* 1](#_Toc507401063)

[*1.2 Product Functions* 1](#_Toc507401064)

[*1.3 Operating Environment* 2](#_Toc507401065)

[*1.4 Design and implementation constraint* 2](#_Toc507401066)

[*1.5 Technologies* 2](#_Toc507401067)

[2. Interfaces 2](#_Toc507401068)

[*2.1* *Use Case Model* 3](#_Toc507401069)

[*2.2* *Sequence Diagram for action Rent* 3](#_Toc507401070)

[*2.3* *Typical course of events for action Rent* 4](#_Toc507401071)

[*2.4* *UML diagram* 5](#_Toc507401072)

[3. The software 5](#_Toc507401073)

[*3.1* *Software database* 5](#_Toc507401074)

[*3.2* *Screenshots from the running software* 6](#_Toc507401075)

[4. System features 8](#_Toc507401076)

[5. Other Nonfunctional Requirements 8](#_Toc507401077)

6. Conclusions..............................................................................................................................10

[References 11](#_Toc507401078)

# LIST OF TABLES

**[Table 1: Course event for action rent](#_Toc506463675)** [4](#_Toc506463675)  
**Table 2: Register table from Database………………………………………………………….5**

**Table 3: Bicycle table from Database……………………………………………………………5**

# LIST OF FIGURES

**Fig 21: Use Case……………………………………………………………………….…………3**

**Fig 3: UML Diagram……………………………………………………………………………5**

**Fig 4: Sequence Diagram…………………………………………………….…………………5**

**Fig 5: Sign Up Page…………………………………………………………………..…………6**

**Fig 6: Register Page……………………………………………………………………..………6**

**Fig 7: User Main Page…………………………………………………………………...………7**

**Fig 8: Admin Manage Users Page………………………………………………………………7**

# Software Requirements

Only in the recent years, in Tirana, a bike sharing activity took place and started to grow larger. Being the only program of its kind and still considered new, it still operates without the need of technology. But analyzing the potential it has to grow larger it is necessary to be updated and digitalized. Bike sharing consists on people renting a bicycle for a limited amount of time and paying a fee after returning it to its destination. One of the biggest problems of the way this bike sharing activity works is the way how users are registered. Currently speaking, when someone wants to rent a bicycle, they show one the employees their ID card. This ID number is written on a file and the ID card is returned to the owner only after they return the bike. This is one of the main reasons that motivated me to work on a simple version of Bike Sharing System. This version is yet to be updated and perfected but its main goal is to help both users and employees on the process of registration and renting the bikes.

## *1.1 System purpose*

Bike Sharing System is designed to help users check for available bicycles, their current location and rent (or reserve) one online. They can also check the amount of money they have to pay for the time period they are going to use it.  
At the other hand the program helps Administrator to manage all users and bicycles. He can update, delete or add new users or available bicycles at any time.

## *1.2 Product Functions*

Bike Sharing System has these important features:

1. Log in as a user or as an administrator: After a successful registration users can directly sign in at their account and start using the program. Meanwhile administrator doesn't have to go through the registration phase because he is already registered in the database and can access the program only through a specific username and password.
2. Manage users and bicycles: This can only be performed from the administrator. He can update, delete or add a new user or available bicycle.
3. Rent Service: Users can choose an available bicycle and rent it for a specific amount of time. Right before renting it they are provided with the amount of money they should pay for the service
4. Search service: Both user and administrator can use the search service. Users can search for any available bicycle by adding a desired location. Meanwhile administrator can search for any user or bicycle by adding any specific keyword.

## *1.3 Operating Environment*

This software is a desktop application designed to work on Windows or Linux Server. Servers should have a distinct room to be placed in.

## *1.4 Design and implementation constraint*

## *1.5 Technologies*

* MySQL: using Wamp Server 64
* Netbeans IDE: Java editor
* Java Swing: for the GUI

The software will be available in English. The administrator will be responsible for the security of the software and he will manage all the data stored in the database. Users cannot intervene into other users’ data. The user that is using an unavailable bicycle won’t appear to other users, only the bicycle status will be visible. The amount of money users have to pay will be calculated only after they choose the amount of time they are going to use the bike. The currency used is ALL

# Interfaces

This application is designed to have multiple interfaces (Graphical Interfaces) in accordance with its multiple users and functions. The 4 main interfaces will be:

* Login and Register Graphical Interfaces
* User Graphical Interface
* Admin Manage Users Graphical Interface
* Admin Manage Bicycles Graphical Interface

These graphical interfaces make the software appear more user-friendly and make it easier to use.

## *Use Case Model*

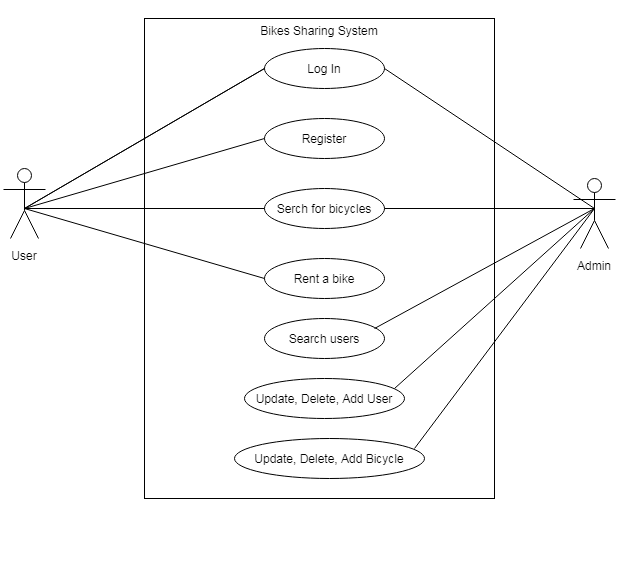


Fig : Use Case

## *Sequence Diagram for action Rent*

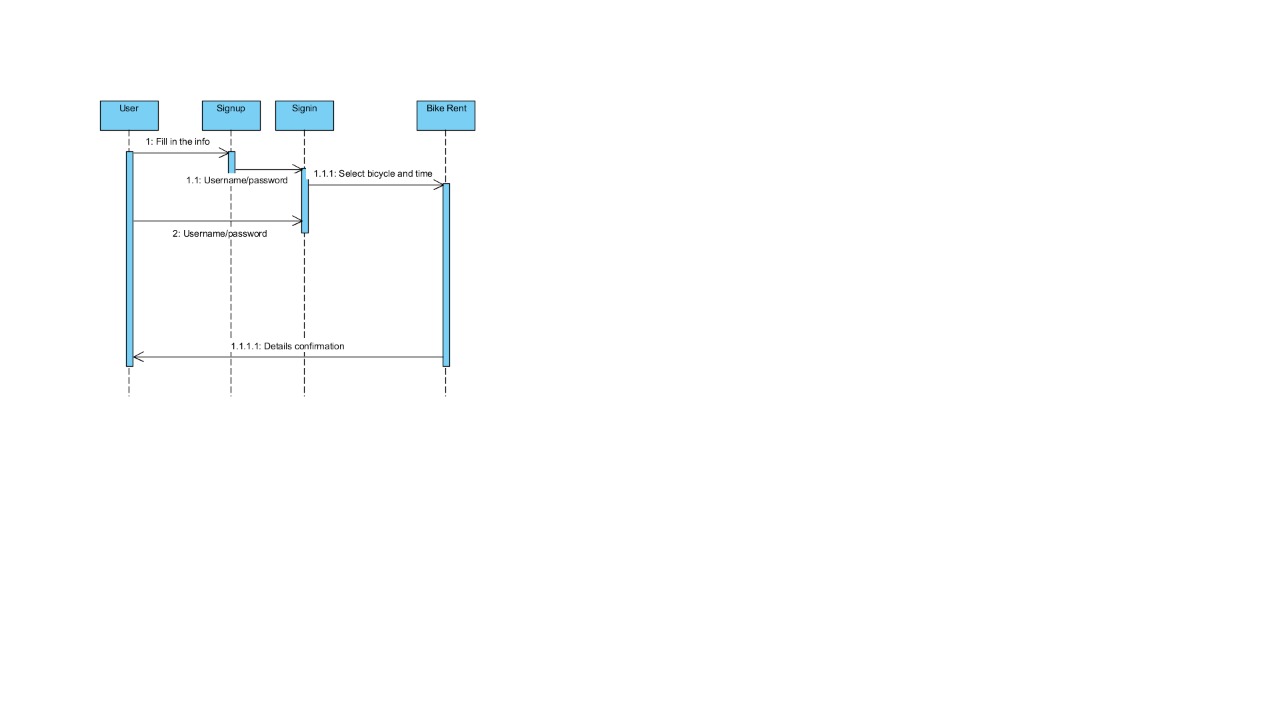


Fig : Sequence diagram

## *Typical course of events for action Rent*

**System:** Bike Sharing System  
**Use Case:** Rent bicycle  
**Actors:** User  
**Purpose:** To search the system for a bike and rent it.  
**Overview:** The user signs in to his account. He searches or clicks on the desired bicycle he wants to rent. He selects the amount of time he will be using the bicycle. The amount of money he has to pay immediately is shown on the screen. A message telling the user that the bike was successfully rent pops up. The user signs out.

Table 1: Course event for action rent

|  |  |
| --- | --- |
| **Actor action** | **System action** |
| User opens the software | A sign in page shows up |
| User signs in to his account | Users main page shows up with a list of all bicycles |
| User selects or searches an available bicycle and the amount of time he will be using it | A message pops up telling the user if this action was successful or not and also the calculated payment is shown on the screen |
| User signs out |  |

## *UML diagram*

## 

Fig : UML Diagram

# The software

## *Software database*

Table 2: Register table from Database

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Id | Name | Last name | Email | Username | Password | Id\_number |

Table 3: Bicycle table from Database

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Id | Number plate | Location | Available | Id\_user |

## *Screenshots from the running software*

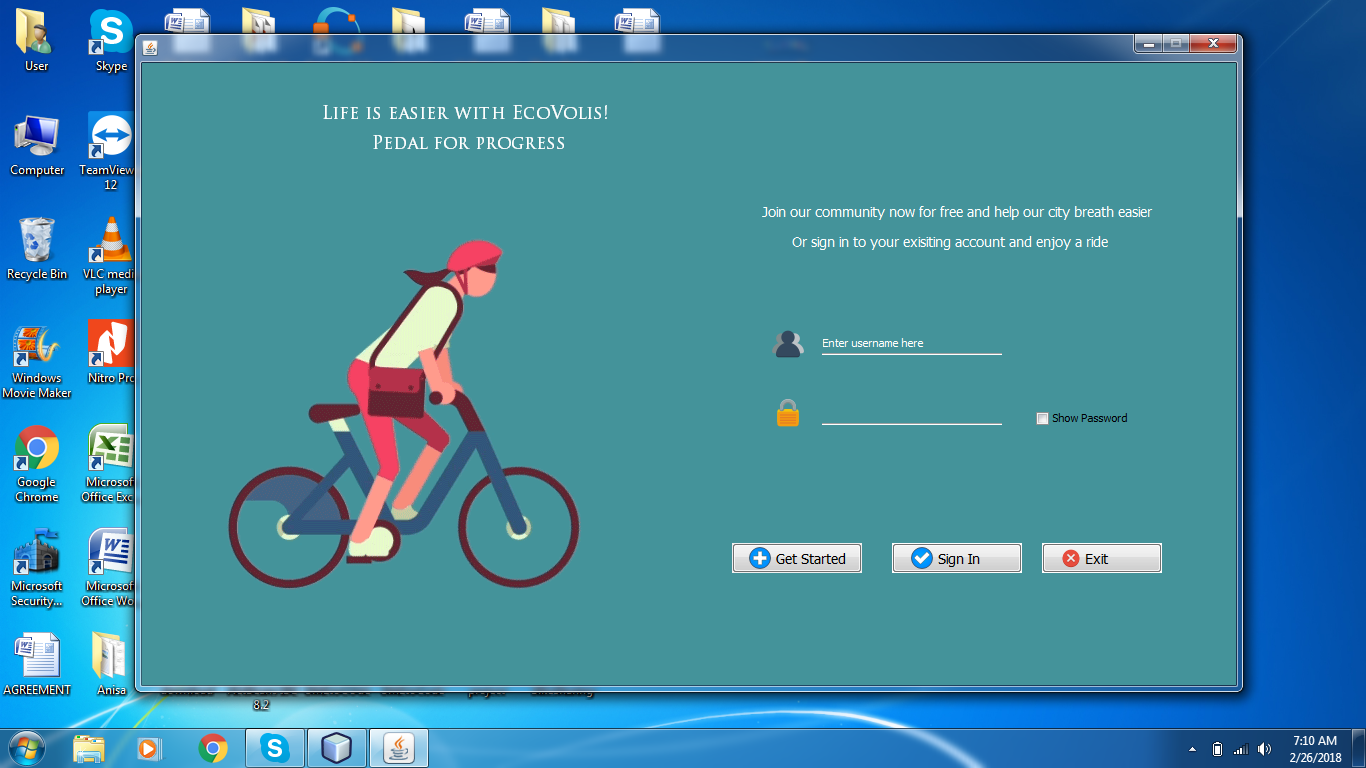
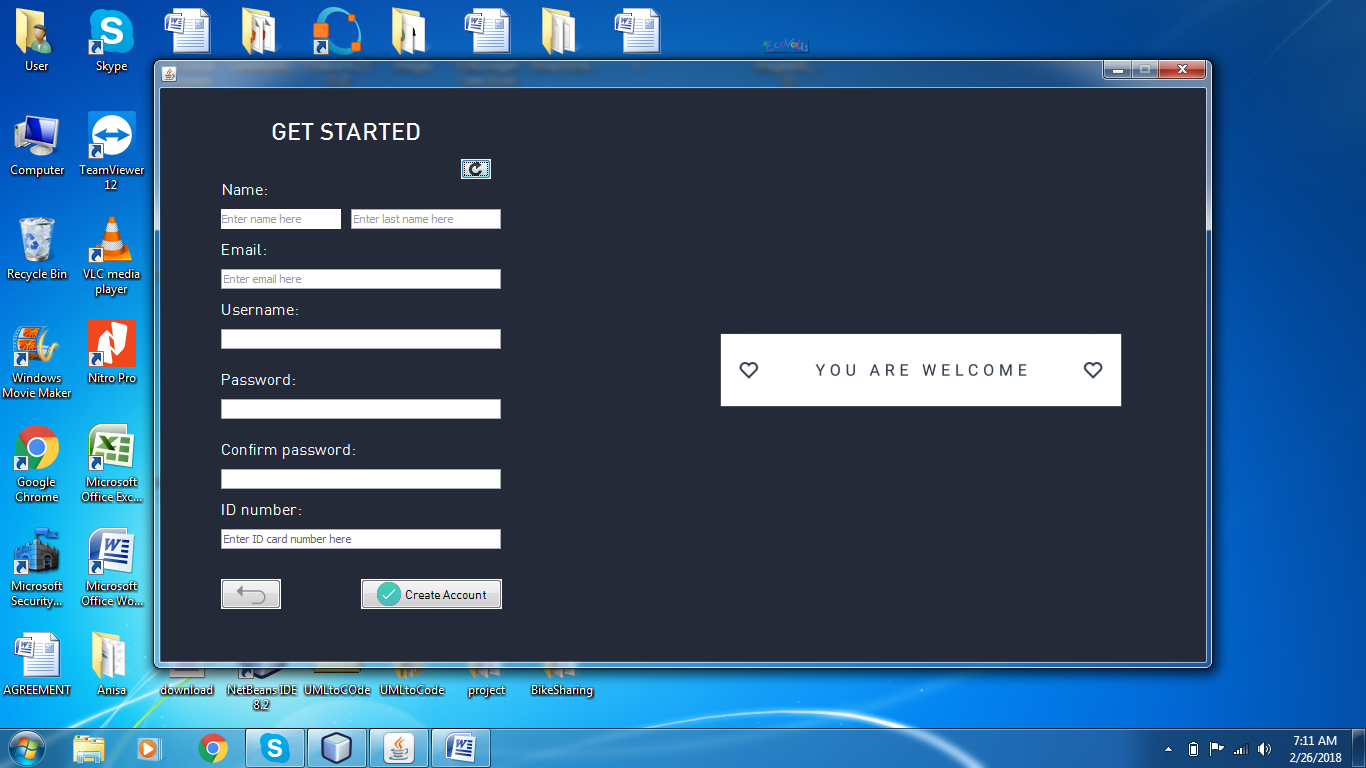


Fig : Sign in page

Fig : Register Page

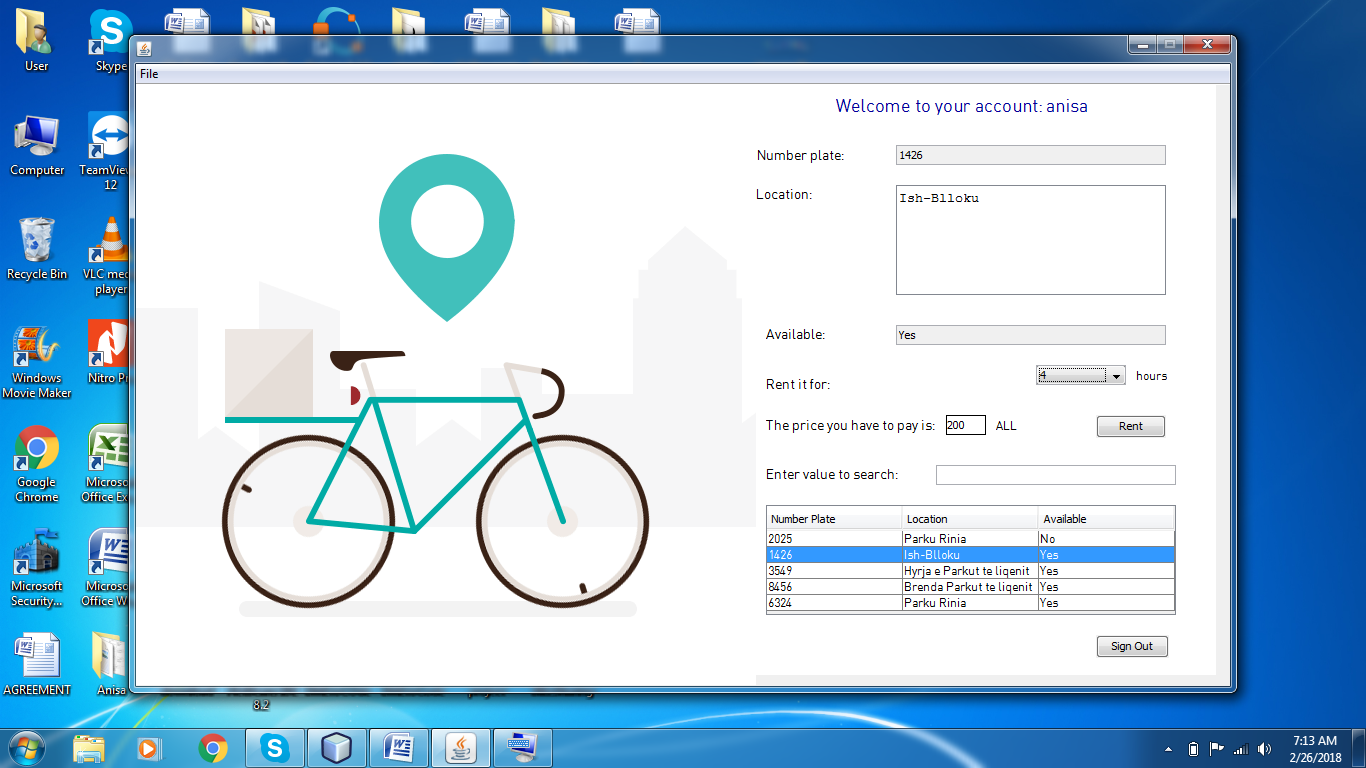
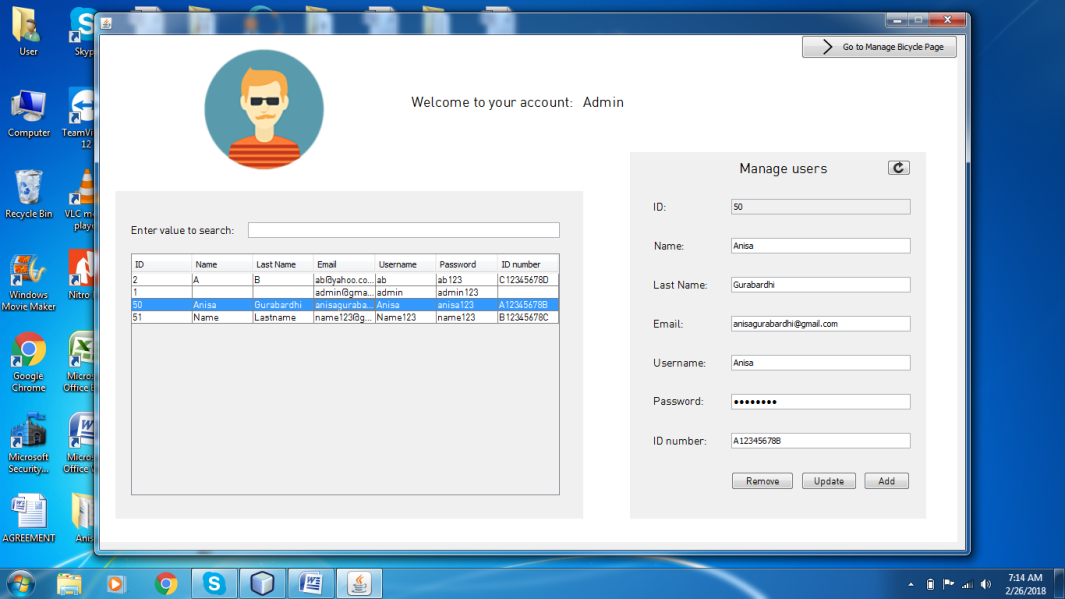


Fig : Admin manage users page

Fig : User main page

# System features

The software is designed to be very easy on use and very functional for both users and administrator.

* 1. ***OS Compatibility***

The System is designed to run on desktop operating system.

* 1. ***Description and priority***

Nowadays, where there are so many operating systems in use, a common solution has to be found. The OOP of the Bike Sharing System, JAVA, can work on the majority of the OS. This feature gives the software the opportunity to work on the majority of the devices used by the clients or the business devices. For the database I have used WampServer 64.

* 1. ***Functional Requirements***

To be fully functional the software just needs a device (pc server) which runs on a Windows, Linux or a Linux based OS. These are the software capabilities that must be present in order for the system to work and the user to carry out the services provided by the HMS.

* 1. ***Multi User Compatibility***

Multiple distinct functions of the systems need to have multiple users.

4.4.1 **Description and Priority**

The software is designed to have many functions that will be needed to best manage the renting process. For this reason, it needs to have only an admin and many users which will profit from the services.

# Other Nonfunctional Requirements

* 1. ***Performance Requirements***

To work and be on full capacity the system and the operating environment where it will work needs to fulfill some conditions. Server has to be secured and has its own private place. It has to be connected with internet 24/7 and an administrator has to keep its well-being.

* 1. ***Security Requirements***

To be fully secure the server on which the system will be installed will be accessed only by administrator of the software. Administrator will be responsible for the well-being and the data protection of the software. We will not be responsible for any issue coming from manual interventions on the server or security breach of the system if the policy defined from us and accepted from you if violated.

# 5. Conclusions

As stated at the introduction of this documentation the target of this project is people who rent bicycles and the employees who work there. This project only shows some core functionalities designed in a simple graphical user interface.

This is yet a simple version of the software and needs to be upgraded and perfected more but still it solves some of the problems that this activity is facing nowadays. One of the problems still not figured out is the way how to get and register the time at which a user rents a bikes. One of the main advantages of this software is the use of a database which will facilitate the employees’ work. It can also attract more users because it makes it easier and more convenient for them to rent a bicycle. This software doesn’t require a lot of memory space and this makes it more comfortable to install. On some other version of the application a renting time might be registered and also a GPS connection to serve the purpose of security.

# References

|  |  |
| --- | --- |
| [1] | Paul Deitel and Harvey Deitel, "Java How to Program", 9th ed, 2011 |
| [2] | Bruce Eckel, "Thinking in Java", 4th ed, 2007. |
| [3]  [4] | MySQL Mother Site @ [www.mysql.com](http://www.mysql.com/).  Arlow, Jim and Ila Neustadt (2002), UML and the Unified Process, Addison Wesley |
|  |  |