**BIKE RENTAL MANAGEMENT SYSTEM**

A Mini Project Report submitted by

### SABIRA V (MES21MCA-2042)

##### to

the APJ Abdul Kalam Technological University

in partial fulfillment of the requirements for the award of the Degree of

Master of Computer Applications



### Department of Computer Applications

MES College of Engineering Kuttippuram, Malappuram - 679582

#### November 2022

**DECLARATION**

I undersigned hereby declare that the project report **BIKE RENTAL MANAGEMENT SYSTEM**, submitted for partial fulfillment of the requirements for the award of degree of Master of Computer Applications of the APJ Abdul Kalam Technological University, Kerala, is a bonafide work done by me under supervision of Ms Priya J.D, Assistant Professor, Department of Computer Applications. This submission represents my ideas in my own words and where ideas or words of others have been included. I have adequately and accurately cited and referenced the original sources. I also declare that I have adhered to ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in my submission. I understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not been previously formed the basis for the award of any degree, diploma or similar title of any other University.

Place:Kuttippuram Date:25/11/2022

SABIRA V

(MES21MCA-2042)



#### DEPARTMENT OF COMPUTER APPLICATIONS MES COLLEGE OF ENGINEERING, KUTTIPPURAM



CERTIFICATE

This is to certify that the report entitled **BIKE RENTAL MANAGEMENT SYSTEM** is a bonafide record of the Mini Project work during the year 2022-23 carried out by **SABIRA V (MES21MCA-2042)** submitted to the APJ Abdul Kalam Technological University, in partial fulfillment of the requirements for the award of the Master of Computer Applications, under my guidance and supervision. This report in any form has not been submitted to any other University or Institution for any purpose.

Internal Supervisor(s) Head of the Department



# Acknowledgements

At the very outset I would like to thank the almighty’s mercy towards me over the years. I wish to express my sincere thanks to my project cordinator, Ms Febin Aziz, Assistant professor, Dept. of Masters in Computer Applications who gave the right information at right time to me in the successful completion of this project. I also thank her for valuable suggestions, guidance, constant encouragement, boundless corporation, constructive comments and motivation extended to me for completion of this project work.

I would express my sincere thanks to my internal guide Ms Priya J.D, Assistant professor in the department for his immense guidance to complete the project successfully.

I would like to express my sincere thanks to all the faculty members of Master of Computer Applications department for their support and valuable suggestion for doing the project work. Last but not least my graceful thanks to my parents, friends and also the persons who supported me directly and indirectly during the project.

##### SABIRA V

##### (MES21MCA-2042)



# Abstract

The project “Bike Rental System” is designed to help the customers to take bikes or two-wheelers for rent. When we go on any trip outside the town or country, we want to be free of time so instead of going through metros and taxis, we prefer to have our own vehicle for rent.

Using this system customers who want to take bikes on rent can register themselves as renters and can take any bike for rent. Addresses of both are required as the customer can only take the bike by going to the address provided and the vehicle owners can know the address that a customer is verified or not.

The customer also has to upload some proofs to take the bike on rent. Proofs like license, pan card, and identity card are compulsory so that no one could run taking the bike. Any customer whose proofs are not uploaded and are not valid will not be allowed to take any bike on rent. This has one admin account that verifies the registering user and one type of user account.

**Contents**

**Declaration 1**

**Certificate 2**

**Acknowledgements 3**

**Abstract 4**

**Contents 5**

**List of Figures 7**

**List of Tables 8**

**1 Introduction 9**

1.1 Background 9

1.1.1Motivation 9

1.2 Objective 9

1.3 Contribution 10

1.4 Report Organization 10

**2 Existing System** 11

**3 Methodology 12**

3.1 Introduction 12

3.2 Product Backlog 14

3.3 User Story 15

3.4 Project Plan 16

3.5 Sprint Backlog (Plan) 17

3.6 Sprint Backlog (Actual) 19

3.7 Database and User Interfaces 20

3.8 Git 20

**4 Conclusion** 21

**Appendix**

Source Code 22

Database Design 27

Git History 34

Screenshots 35

# List of Figures

* 1. [Dataflow Diagram](#_26in1rg) 31
  2. ER Diagram 33
  3. [Project Plan](#_lnxbz9) 16
  4. [Product Backlog](#_35nkun2) 14
  5. [Sprint Backlog 1](#_1ksv4uv) 17
  6. [Sprint Backlog 2](#_44sinio) 18
  7. [Sprint Backlog 3](#_2jxsxqh) 18
  8. [Sprint Backlog Actual 1](#_z337ya) 19
  9. [Sprint Backlog Actual 2](#_3j2qqm3) 19
  10. [Sprint Backlog Actual 3](#_1y810tw) 20



**List of Tables**

1. Bike 27
2. Complaint 27
3. Login 28
4. Notification 28
5. Payment 28
6. Request 29
7. Staff 29
8. User 30

# Chapter 1 Introduction

## 1.1 Background

A bike rental is a rental vehicle that can be used temporarily for a fee during a specified period. Getting a rental bike helps people get around despite the fact they do not have access to their own personal vehicle or don't own a vehicle at all. The individual who needs a bike must contact a rental bike company and contract out for a vehicle. This system increases customer retention and simplifies vehicle and staff management.

### 1.1.1 MOTIVATION

The present system is a manual system that is used to know about the vehicles that are available for renting. This system is time consuming. To overcome this we have come up with an idea. That is we build a bike rental management system. It will be very easy for the customers to take two wheeler for rent using a digital system.

## 1.2 Objective

To produce a web-based system that allows the customer to register and reserve bikes online and for the company to effectively manage their bike rental business. To ease customer’s tasks whenever they need to rent a bike. To ensure availability of vehicle round the clock. Maintaining record of each booking history . Online payment option. Login and account generation.

## 

## 1.3 Contribution

The major contributions in this projects are:

1. Designed and developed a new system for renting two wheelers.

2. Designed and developed a generic product that any organization can utilize this system.

3. Proper security measures are introduced in this system.

4. The system is developed as a web application which provides any where any time access.

5. This system can run in any Operating System.

6. Developed a solution for issues in the manual system for renting two wheelers

## 1.4 Report Organization



The project report is divided into four sections. Section 2 describes Existing System. Section 3 describes the methodology used for implementing the project. Section 4 gives the conclusion.

**Chapter 2**

**Existing System**

The current structure Bike rental management System is :

1. It is a direct system
2. Two wheelers is renting manually
3. Time consuming
4. High workload

When we look around the existing system, it have many drawbacks like high workload,

time consuming and difficulty of analysing the details of bikes manually. It’s very difficult for the customers to select the bike in affordable price manually and also difficult to ensure the conditions are satisfiable for a ride. It’s also difficult for number of customers to take bike at the same time because the existing system is direct renting from vehicle shop.

By considering all these factors, the requirement system to be developed for implementation

of the Bike Rental Management System to ensure all customers can know and take bikes for their needs quickly.

As a solution the Bike Rental Management System is prepared, which

solve all the problems in the existing system. And some more improvement are also incorporated with it. The methodology followed are described in Chapter 3.



# Chapter 3 Methodology

## 3.1 Introduction

This project is done by agile methodology. It is found that agile model of software development is suitable and is the best method for the development of this system. Agile methodology mainly focused on the client satisfaction through continuous delivery. Also it sets a minimum number of requirements and turns them in to a deliverable product. As this project has many individual requirements which can be delivered in parts and the user can gradually improve their work efficiency. Agile methodology has a family of methods of which scrum is selected for the development of this project. Scrum is process framework that has been used to manage complex product development. It is not a process or technique for building products rather it isa framework within which various processes can be employed. Also it is suitable method to support the development process. It focuses on lean software development and has in building better software effectively and efficiently.

Agile is one of the most widely used and recognized software development framework.

The methodology those experts agreed upon was described as ‘lightweight’ and fast. Agile

is also about being the adaptive and continuous improvement, as much as it is about constant

feedback and speed of delivery. “Agile is a software development approach where a self-sufficient and cross-functional team works on making continuous deliveries through iterations and evolves throughout the process by gathering feedback from the end users.

The major rules in scrum methodology are

**1.** **The product owner (PO) :** Who represents the stakeholder and the business.

**2. The scrum master :** Ensures the process followed, removes obstructions, and protects the development system

**3. Development team:** Cross functional, self organizing team who actually do the actual analysis, design implementation and testing process.

They work together in iterative time boxed duration called sprints. The first step is the creation of the product backlog by the PO. It’s a to-do list of stuff to be done by the scrum team.Then the scrum team selects the top priority items and tries to finish them within the time box called a sprint. An easier way to remember all of this is to memorize the 3-3-5 framework. It means that a scrum project has 3 roles, 3 artifacts, and 5 events

These are:-

1. Roles : Product Owner, Scrum Master, and development team.

2. Artifacts : Product Backlog, Sprint Backlog and Product Increment.

3. Events : Sprint, Sprint planning, Daily Scrum, Sprint review and Sprint retrospective

The framework begins with a simple premise start with what can be seen or known. After that

the progress is tracked and tweak as necessary. The three pillars of scrum are transparency, inspection and adaptation. In scrum everyone has a role.

Bike Rental Management System is implemented using Django-Python,a framework of python language.Django is a high level Python web framework that encourages rapid development and clean,pragmatic design and it’s great for creating database driven website.it’s free and open sourse

The Git is used as the version control system for this project. Version control is a system

that records changes to a file or set of files over time so that a specific versions can be recalled later. Version control systems are a category of software tools that help a software team for managing changes to source code over time. Version control software keeps track of every modification to the code in a special kind of database. If a mistake is made, developers can turn back the clock and compare earlier versions of the code to help fix the mistake while

minimizing disruption to all team members.

**3.2 Product Backlog**

A product backlog is a list of the new features, changes to existing features, bug fixes, infrastructure changes or other activities that a team may deliver in order to achieve a specific

outcome.The product backlog is the single authoritative source for things that a team works on. That means that nothing gets done that isn’t on the product backlog. Conversely, the presence of a product backlog item on a product backlog does not guarantee that it will be delivered. It represents an option the team has for delivering a specific outcome rather than a commitment. It should be cheap and fast to add a product backlog item to the product backlog, and it should be equally as easy to remove a product backlog item that does not result in direct progress to achieving the desired outcome or enable progress toward the outcome. The Scrum Product Backlog is simply a list of all things that needs to be done within the project. It replaces the traditional requirements specification artifacts. These items can have a technical nature or can be user-centric e.g. in the form of user stories.The product backlog of the system is given in Table below

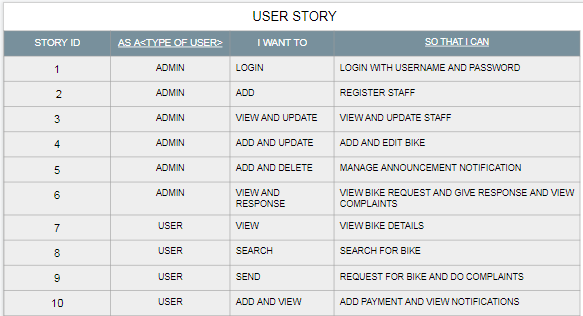
| ID | NAME | PRIORITY | ESTIMATE  <HOURS> | STATUS<planned/in progress/completed> |
| --- | --- | --- | --- | --- |
| 1 | REGISTRATION | HIGH | 5 | COMPLETED |
| 2 | LOGIN | HIGH | 5 | COMPLETED |
| 3 | CRUD OPERATIONS  (BIKE,STAFF,USER) | HIGH | 20 | COMPLETED |
| 4 | REQUEST AND RESPONSE (BIKE) | HIGH | 15 | COMPLETED |
| 5 | NOTIFICATION,COMPLAINT,PAYMENT | MEDIUM | 5 | COMPLETED |

**3.3 User Story**

A key component of agile software development is putting people first, and user-stories put

actual end users at the center of the conversation. Stories use non-technical language to provide context for the development team and their efforts. After reading a user story, the team knows why they are building what they’re building and what value it creates. A user

story is a tool used in agile software development to capture a description of a software feature from an end-user perspective. The user story describes the type of user, what they want and why. A user story helps to create a simplified description of a requirement. User stories are one of the core components of an agile program. They help provide a user-focused framework for daily work — which drives collaboration, creativity, and a better product overall. The user story of system is given in Table





**3.4 Project Plan**

A project plan that has a series of tasks laid out for the entire project, listing task durations,

responsibility assignments, and dependencies. Plans are developed in this manner based on

the assumption that the Project Manager, hopefully along with the team, can predict up front

everything that will need to happen in the project, how long it will take, and who will be able

to do it. Project plan is given in Table

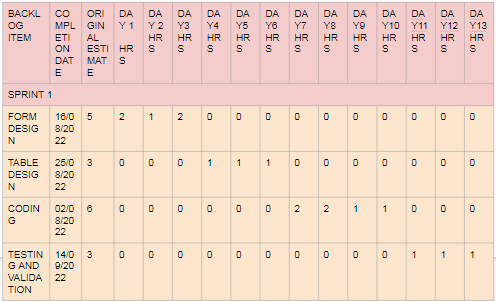
| USER STORY ID | SPRINT | START DATE | END DATE | HOURS | STATUS |
| --- | --- | --- | --- | --- | --- |
| 1 | SPRINT 1 | 16/08/22 | 24/08/22 | 17 | COMPLETED |
| 2 | 25/08/22 | 31/08/22 | COMPLETED |
| 3 | 02/09/22 | 14/09/22 | COMPLETED |
| 4 | SPRINT 2 | 15/09/22 | 23/09/22 | 16 | COMPLETED |
| 5 | 26/09/12 | 30/09/22 | COMPLETED |
| 6 | 03/10/22 | 12/10/22 | COMPLETED |
| 7 | SPRINT 3 | 13/10/22 | 18/10/22 | 17 | COMPLETED |
| 8 | 19/10/22 | 24/10/22 | COMPLETED |
| 9 | 25/10/22 | 01/11/22 | COMPLETED |
| 10 | 02/11/22 | 16/11/22 | COMPLETED |

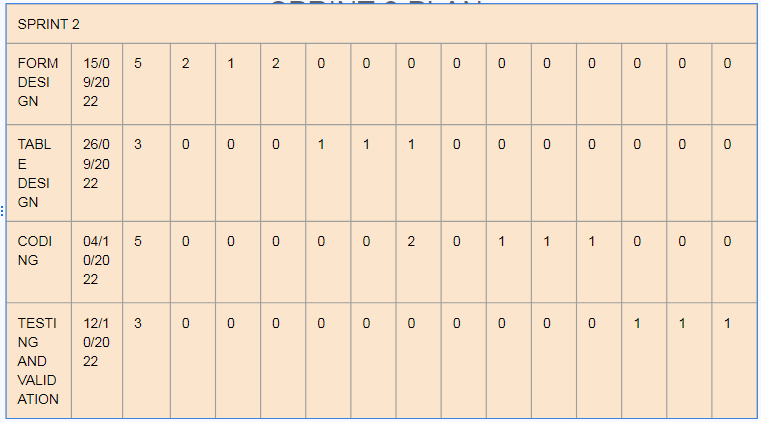
**3.5 Sprint Backlog (Plan)**

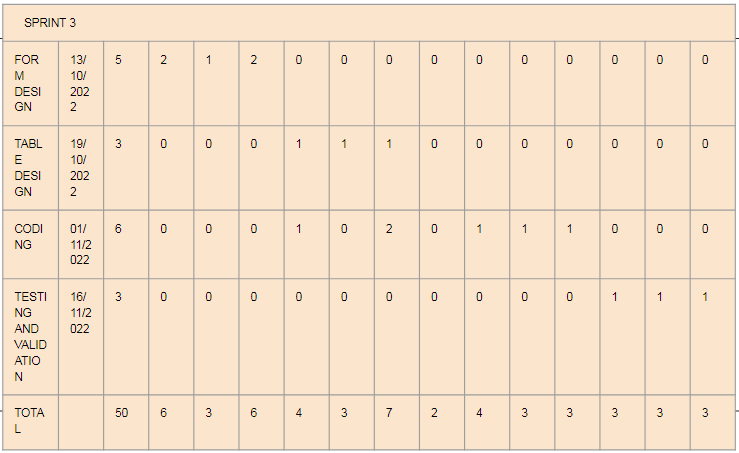
The sprint backlog is a list of tasks identified by the Scrum team to be completed during the

Scrum sprint. During the sprint planning meeting, the team selects some number of product

backlog items, usually in the form of user stories, and identifies the tasks necessary to complete each user story. Most teams also estimate how many hours each task will take someone on the team to complete. Sprint plan of this project is given below.





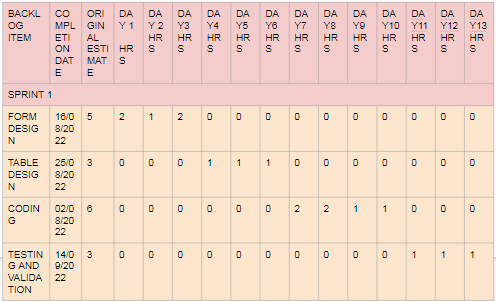


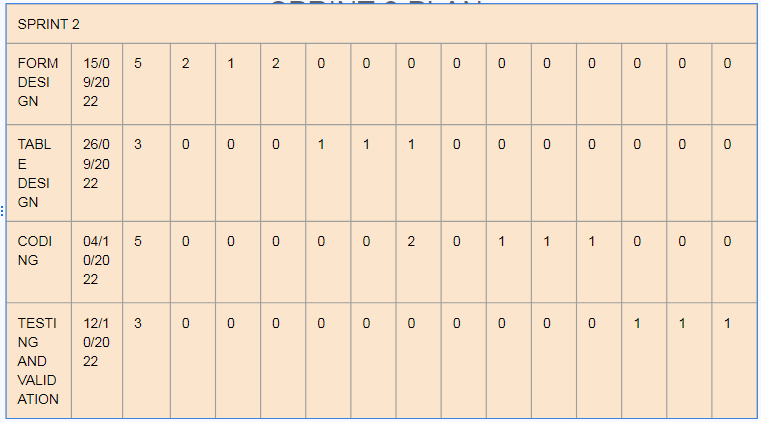
**3.6 Sprint Backlog (Actual)**

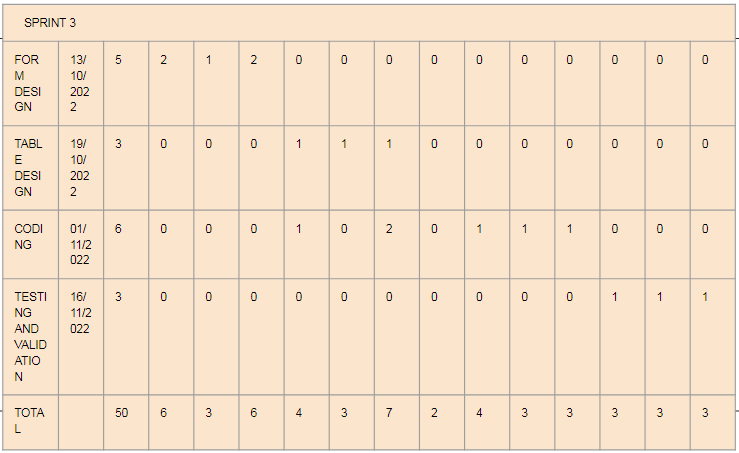
Actual sprint backlog is what adequate sprint planning is actually done by project team there

may or may not be difference in planned sprint backlog. The detailed sprint backlog (Actual)

is given below.







**3.7 Database and User Interfaces**

The data that are captured through this system using different user friendly graphical user interfaces which are shown in Appendix .The data captured are stored in MySQL database and the database description are shown in Appendix which gives the details of the type of data and other characteristics of the data in database design. Sample source code is also attached in Appendix.

**3.8 Git**

Git is a free and open source distributed version control system designed to handle everything

from small to very large projects with speed and efficiency. To show the continuous development of the project the Github histories is shown in Appendix

# Chapter 4 Conclusions

Bike Rental Management System is a web application for two wheeler renting.. It is easy to use and more interactive to the users. The main drawback of the existing result system is that it was a direct system and also it’s very time consuming and high workload. Bike Rental Management System is sub divided into two main subsystems (customer and administrator) that are designed to give the system maximum benefit by demonstrating carefully each subsystem service. The system is easy to use, simple user interface, reduces the time consumption and work load. The project is developed using Django-Python as frond end and MySQL as back end and Pycharm as IDE.



**Appendix**

## Source Code



<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Request</title>

</head>

<body><center>

<form method="post" enctype="multipart/form-data">

{% csrf\_token %}

<table>

<tr>

<td>Name</td>

<td><input type="text" value="{{ u.username }}"></td>

{# <td><select name="ee" required="">#}

{# {% for i in u %}#}

{# <option value="{{ i.u\_id }}">{{ i.username }}</option>#}

{# {% endfor %}#}

{# </select> </td>#}

</tr>

<tr>

<td>Bike Name</td>

<td><input type="text" value="{{ p.name }}"></td>

{# <td><select name="ef" required="">#}

{# {% for i in p %}#}

{# <option value="{{ i.b\_id }}">{{ i.name }}</option>#}

{# {% endfor %}#}

{# </select> </td>#}

</tr>

{# <tr>#}

{# <td>Bike Name</td>#}

{# <td><input type="text" name="bike" value="{{ i.name }}"></td>#}

{# </tr>#}

<tr>

<td>Date</td>

<td><input type="date" name="date" required=""></td>

</tr>

<tr>

<td>Time</td>

<td><input type="time" name="time" required=""> </td>

</tr>

<tr>

<td>Document</td>

<td><input type="file" name="img" required=""></td>

</tr>

<tr>

<td><input type="submit" value="Request"></td>

</tr>

</table>

</form>

</center>

</body>

</html>

{% endblock %}

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>View Request</title>

</head>

<body><center>

<table border="2" cellpadding="2">

<h1>Two wheeler</h1>

<tr>

<th>Date</th>

<th>Time</th>

<th>Document</th>

<th>Username</th>

<th>Bike Name</th>

<th>Status</th>

</tr>

{% for i in x %}

<tr>

<td>{{ i.date }}</td>

<td>{{ i.time }}</td>

<td><img src="/static/{{ i.document }}"></td>

<td>{{ i.u.username }}</td>

<td>{{ i.b.name }}</td>

<td>{{ i.status }}</td>

<td><a href="{% url 'kk' idd=i.r\_id %}" >Approve</a></td>

<td><a href="{% url 'mm' idd=i.r\_id %}" >Reject</a></td>

<td><a href="/static/{{ i.document }}">download</a></td>

</tr>

{% endfor %}

</center>

</table>

</body>

</html>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>View Request</title>

</head>

<body><center>

<table border="2" cellpadding="2">

<h1>Two wheeler</h1>

<tr>

<th>Date</th>

<th>Time</th>

<th>Document</th>

<th>Username</th>

<th>Bike Name</th>

<th>Status</th>

</tr>

{% for i in x %}

<tr>

<td>{{ i.date }}</td>

<td>{{ i.time }}</td>

<td>{{ i.document }}</td>

<td>{{ i.u.username }}</td>

<td>{{ i.b.name }}</td>

<td>{{ i.status }}</td>

<td><a href="/request/pay/{{ i.r\_id }}">Pay</a> </td>

</tr>

{% endfor %}

</center>

</table>

</body>

</html>

from django.shortcuts import render

from pip.\_vendor.six import u

from request.models import Request

from django.core.files.storage import FileSystemStorage

from user.models import User

from bike.models import Bike

from payment.models import Payment

import datetime

# Create your views here.

def post\_request(request,idd):

ss=request.session["uid"]

ob1 = User.objects.get(u\_id=ss)

ob2 = Bike.objects.get(b\_id=idd)

context = {

'u': ob1,

'p': ob2

}

if request.method == "POST":

obj=Request()

obj.date=request.POST.get('date')

obj.time=request.POST.get('time')

# obj.document='1'

myfile=request.FILES['img']

fs=FileSystemStorage()

filename=fs.save(myfile.name,myfile)

obj.document=myfile.name

obj.status='pending'

obj.u\_id=ss

# obj.bikename=request.POST.get('bike')

obj.b\_id=idd

obj.save()

return render(request,'request/request.html',context)

def view\_request(request):

obj=Request.objects.all()

context={

'x':obj

}

return render(request,'request/view.html',context)

def viewstatus(request):

ss = request.session["uid"]

obj = Request.objects.filter(u\_id=ss)

context = {

'x': obj

}

return render(request,'request/viewstatus.html',context)

def approve(request,idd):

obj=Request.objects.get(r\_id=idd)

obj.status='approved'

obj.save()

return view\_request(request)

def reject(request,idd):

obj=Request.objects.get(r\_id=idd)

obj.status='rejected'

obj.save()

return view\_request(request)

def payment(request,idd):

ob1 = User.objects.all()

ob2 = Bike.objects.all()

obj=Request.objects.get(r\_id=idd)

context={

'u': ob1,

'v': ob2,

'x':obj

}

if request.method == 'POST':

obj=Payment()

obj.amount=request.POST.get('amount')

obj.date=datetime.datetime.today()

obj.time=datetime.datetime.now()

obj.u\_id=request.POST.get('ee')

obj.b\_id=request.POST.get('ef')

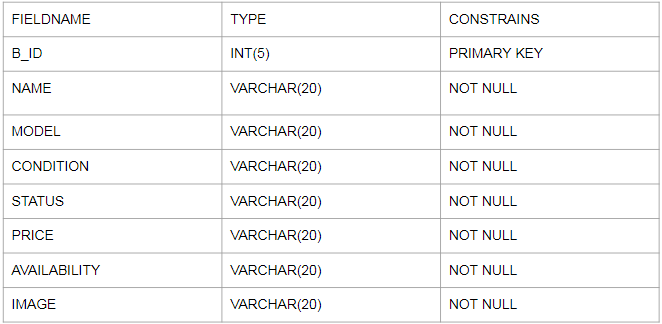
obj.save()

return view\_request(request)

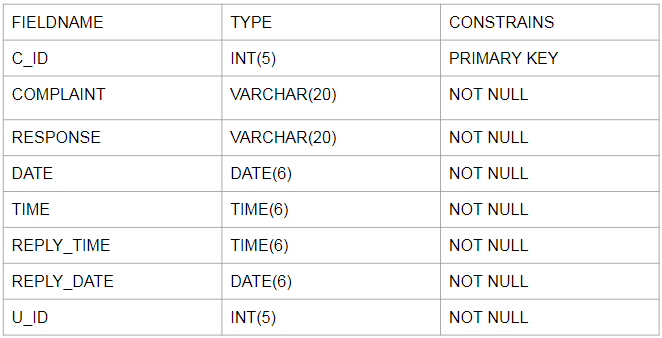
return render(request, 'payment/payment.html',context)

**DataBase Design**

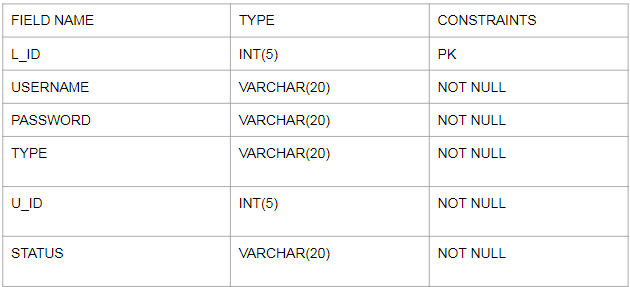
**BIKE**



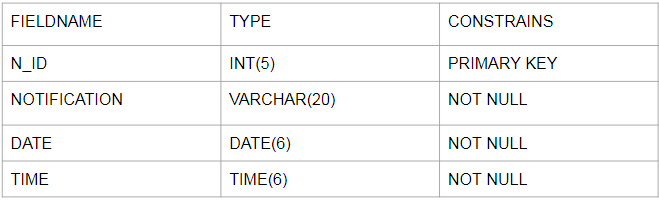
**COMPLAINT**

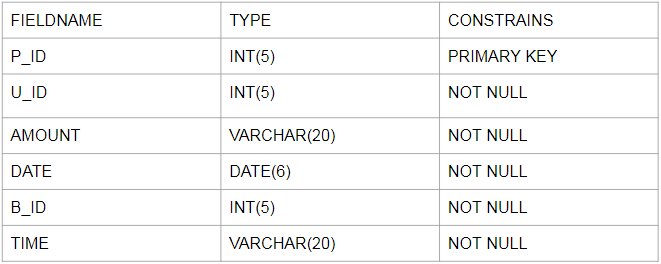


**LOGIN**

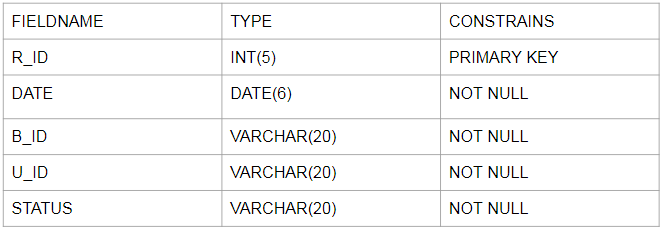


**NOTIFICATION**

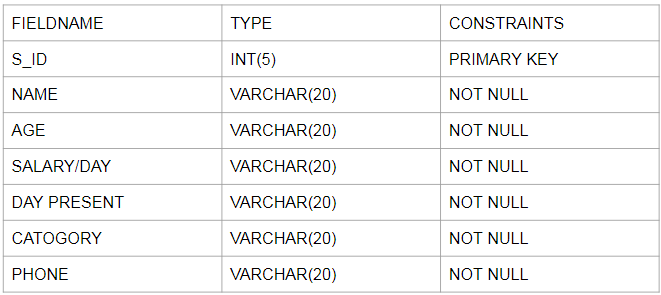


**PAYMENT**

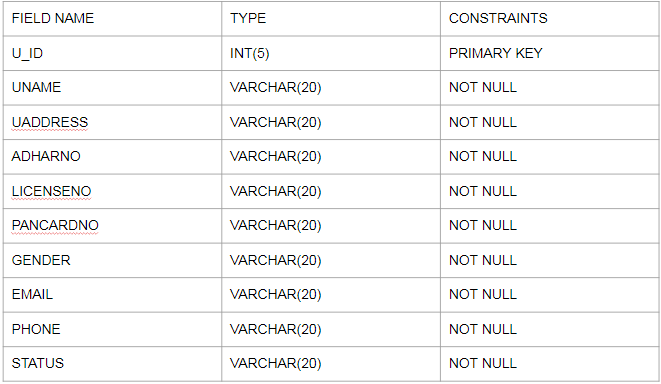
**REQUEST**



**STAFF**

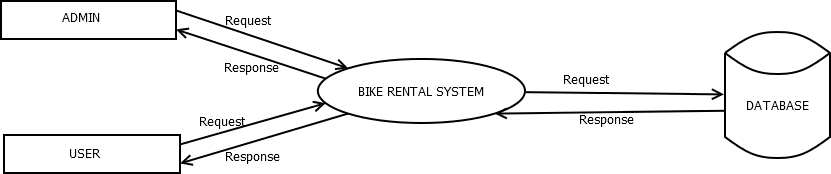


**USER**

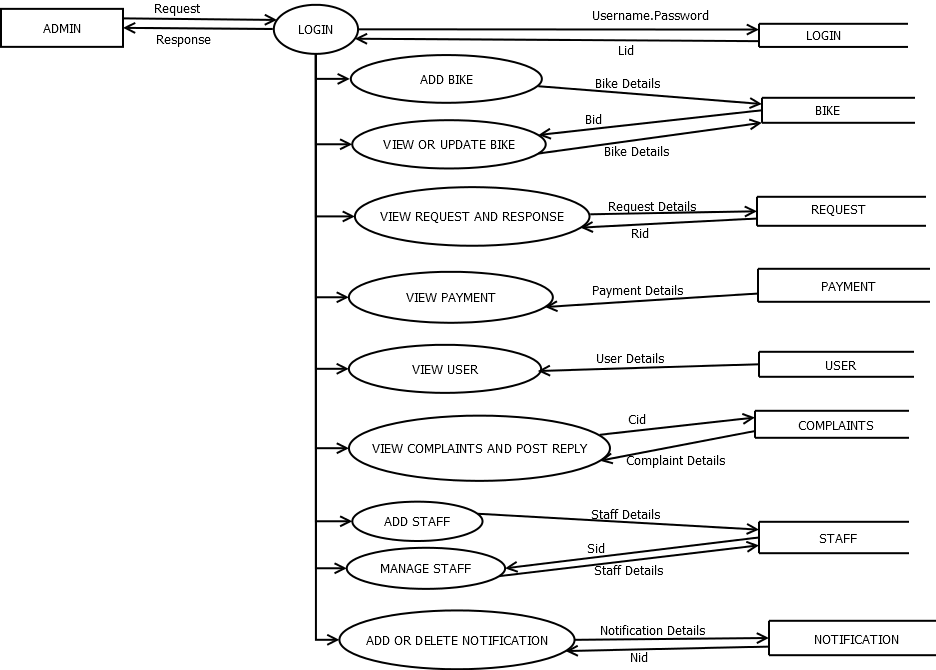


## Data Flow Diagram

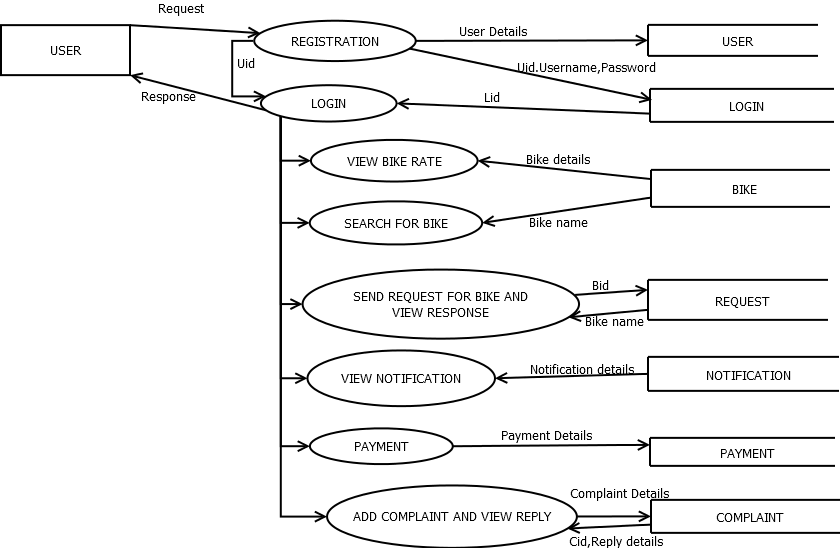
**LEVEL 1**



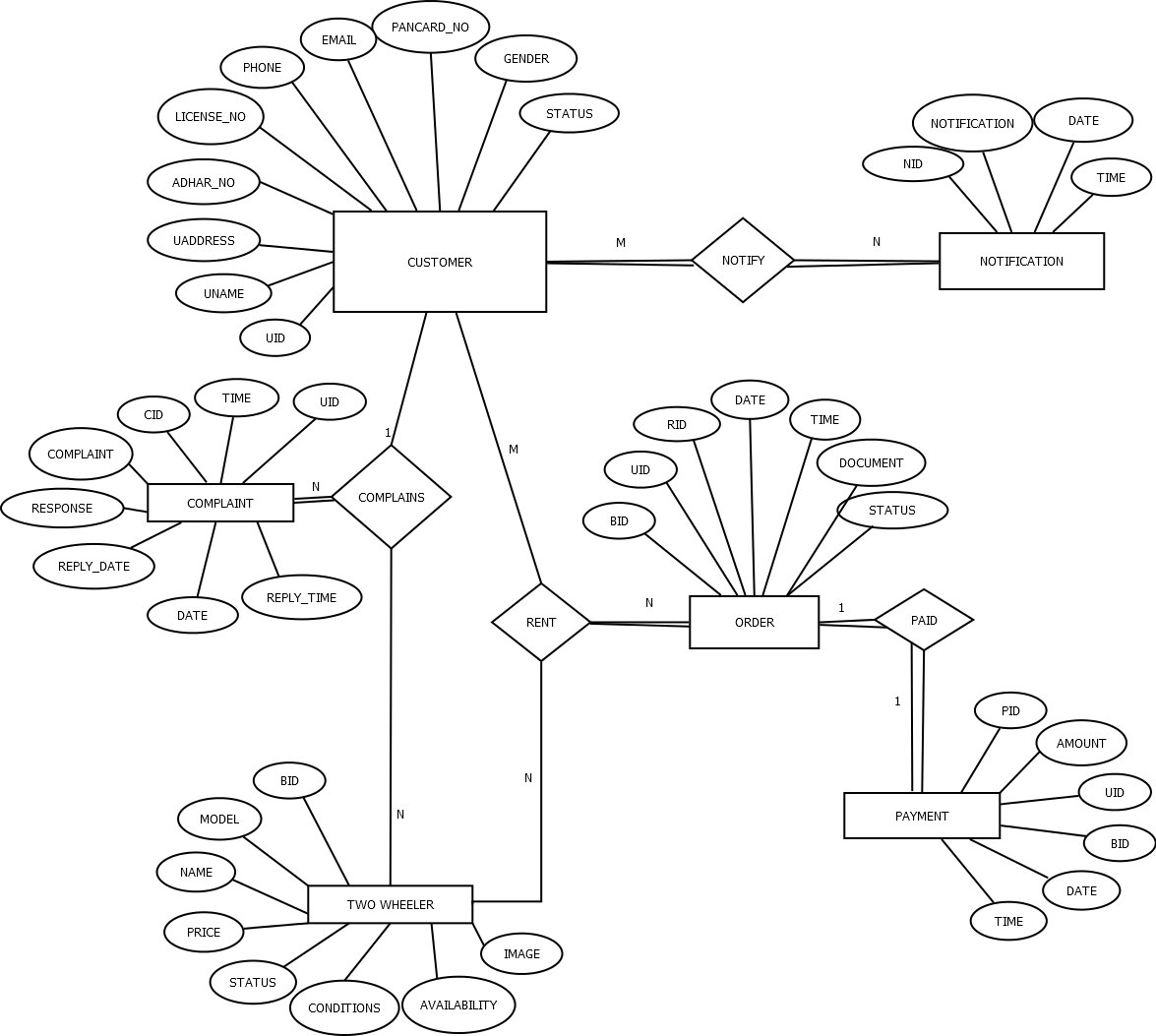
LEVEL 1.1



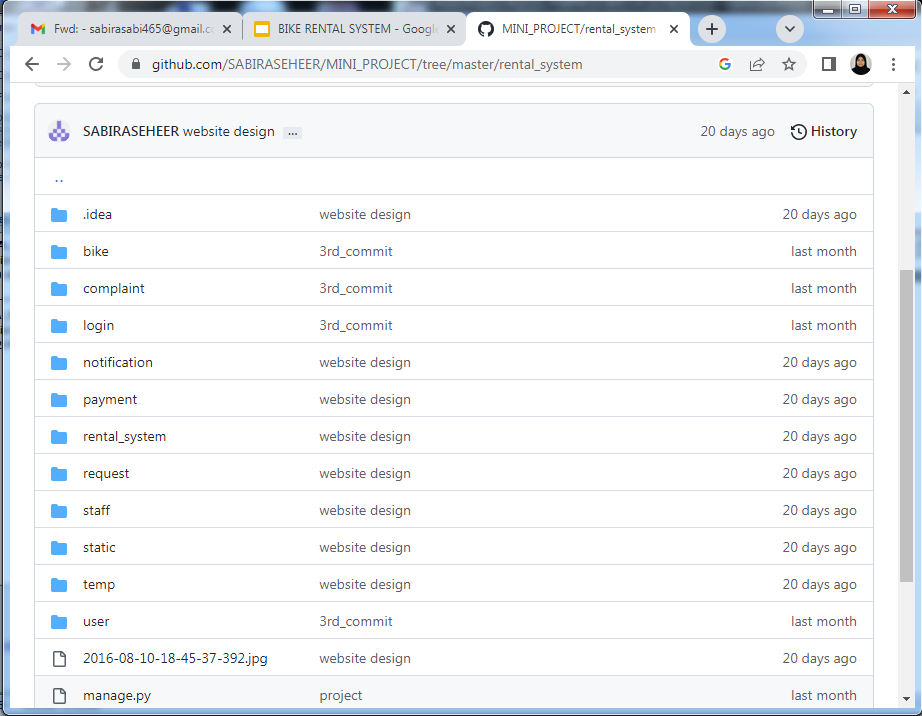
LEVEL 1.2

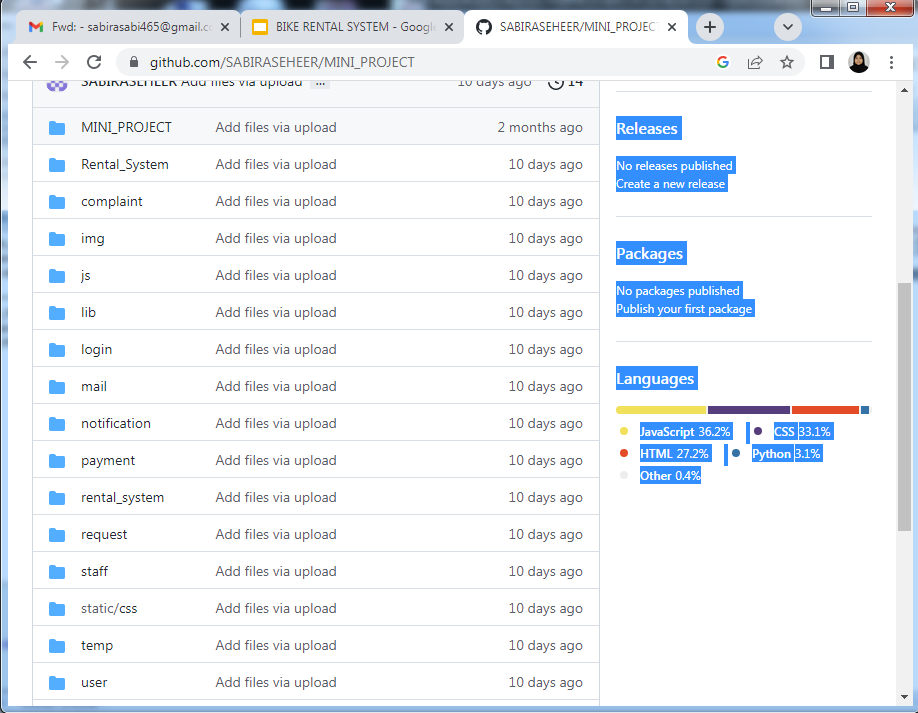


ER Diagram



GIT HISTORY





SCREENSHOTS

