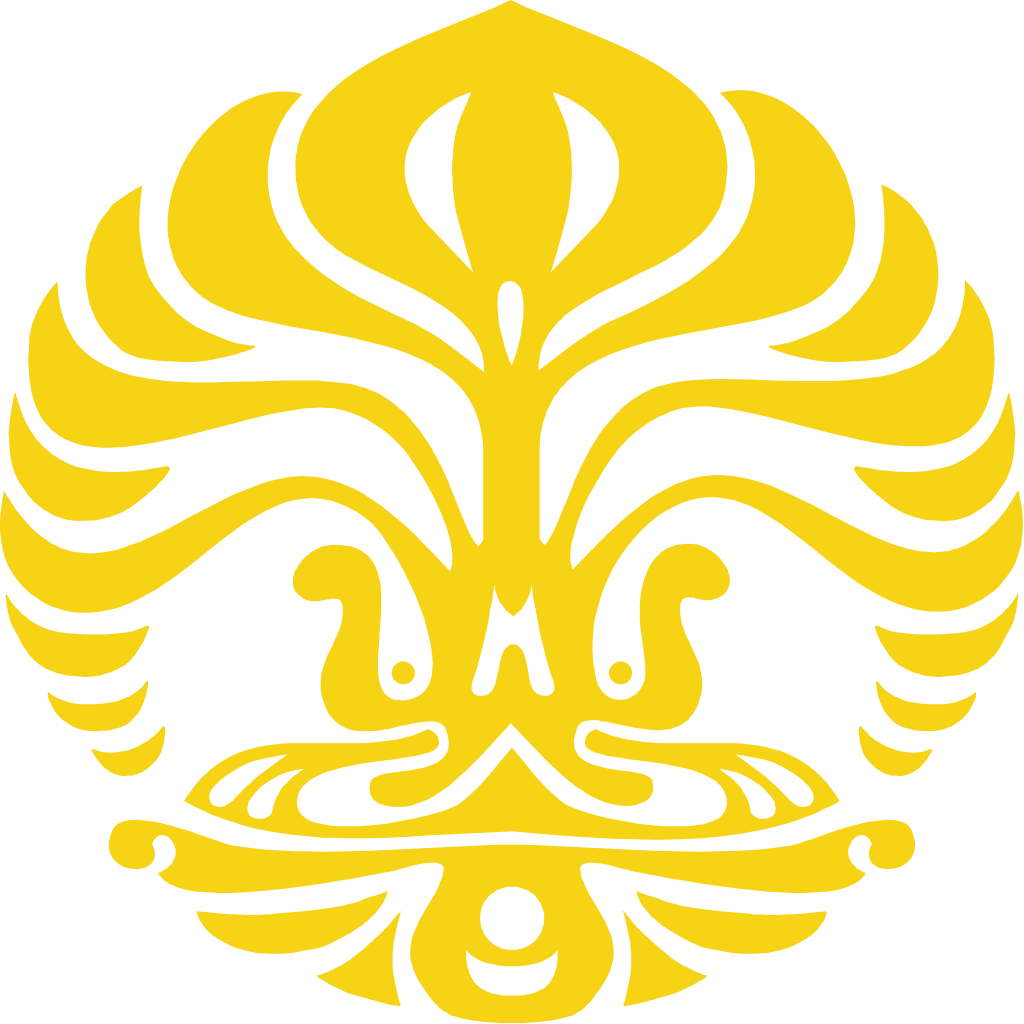
Motorcycle Parking Registration System



Arranged by:

Jauharil Firdaus Bassam 1606995935

Muhamad Luthfi Mufadel 1606827006

Muhammad Wito Malik 1606822844

**Computer Engineering**

**Faculty of Engineering**

**University of Indonesia**

**2018**

Contents

[**Chapter 1** 1](#_Toc533790866)

[**1.1.** **Introduction** 1](#_Toc533790867)

[**1.2.** **Purpose** 1](#_Toc533790868)

[**1.3.** **Target** 1](#_Toc533790869)

[**1.4.** **Tools** 1](#_Toc533790870)

[**1.5.** **Risk Analysis** 2](#_Toc533790871)

[**Chapter 2** 3](#_Toc533790872)

[**2.1.** **Project Integration Management** 3](#_Toc533790874)

[**2.2.** **Project Scope Management** 3](#_Toc533790875)

[**2.3.** **Project Time Management** 3](#_Toc533790876)

[**2.4.** **Project Cost Management** 3](#_Toc533790877)

[**2.5.** **Project Quality Management** 3](#_Toc533790878)

[**2.6.** **Project Human Resources Management** 4](#_Toc533790879)

[**2.7.** **Project Communication Management** 4](#_Toc533790880)

[**2.7.1.** **Information Distribution** 4](#_Toc533790881)

[**2.7.2.** **Information Distribution** 4](#_Toc533790882)

[**2.8.** **Project Risk Management** 4](#_Toc533790883)

[**2.8.1.** **Risk Identification** 5](#_Toc533790884)

[**2.8.2.** **Risk Quantification** 5](#_Toc533790885)

[**2.8.3.** **Risk Response Development** 5](#_Toc533790886)

[**2.8.4.** **Risk Respond** 5](#_Toc533790887)

[**2.9.** **Project Procurement Management** 6](#_Toc533790888)

[**2.9.1 Project Planning** 6](#_Toc533790889)

[**CHAPTER 3 DESIGN DIAGRAMS** 7](#_Toc533790890)

[**3.1.** **Use-Case Diagram** 7](#_Toc533790891)

[**3.2.** **Class Diagram** 8](#_Toc533790892)

[**3.3.** **Object Diagram** 9](#_Toc533790898)

[**3.4.** **Collaboration Diagram** 9](#_Toc533790899)

[**3.5.** **Sequence Diagram** 10](#_Toc533790900)

[**3.6.** **State Diagram** 10](#_Toc533790901)

[**3.7.** **Activity Diagram** 11](#_Toc533790902)

[**3.8.** **Component Diagram** 11](#_Toc533790903)

[**3.9.** **Deployment Diagram** 12](#_Toc533790915)

[**CHAPTER 4 IMPLEMENTATION** 13](#_Toc533790916)

[**4.1.** **Source Code** 13](#_Toc533790921)

[**4.2.** **Implement to Hosting Service** 48](#_Toc533790922)

[**CHAPTER 5 TESTING ANALYSIS** 49](#_Toc533790923)

[**5.1.** **Introduction** 49](#_Toc533790924)

[**5.2.** **Compatibility Testing** 49](#_Toc533790925)

[**5.2.1.** **Test Risk / Issues** 49](#_Toc533790926)

[**5.2.2.** **Items to be Tested** 49](#_Toc533790927)

[**5.2.3.** **Test Approach** 49](#_Toc533790928)

[**5.2.4.** **Test Regulatory / Mandate Criteria** 49](#_Toc533790929)

[**5.2.5.** **Test Pass / Fail Criteria** 49](#_Toc533790930)

[**5.2.6.** **Test Entry / Exit Criteria** 49](#_Toc533790931)

[**5.2.7.** **Test Deliverables** 50](#_Toc533790932)

[**5.2.8.** **Test Suspension** 51](#_Toc533790933)

[**5.2.9.** **Test Environmental** 52](#_Toc533790934)

[**5.3.** **Functional Testing** 52](#_Toc533790935)

[**5.3.1.** **Test Risks / Issues** 52](#_Toc533790936)

[**5.3.2.** **Item to be Tested** 52](#_Toc533790937)

[**5.3.3.** **Test Approach** 53](#_Toc533790938)

[**5.3.4.** **Test Regulatory / Mandate Criteria** 53](#_Toc533790939)

[**5.3.5.** **Test Pass / Fail Criteria** 53](#_Toc533790940)

[**5.3.6.** **Test Entry / Exit Criteria** 53](#_Toc533790941)

[**5.3.7.** **Test Deliverables** 53](#_Toc533790942)

[**5.3.8.** **Test Suspension / Resumption Criteria** 54](#_Toc533790943)

[**5.3.9.** **Test Environmental / Staffing /Training Needs** 54](#_Toc533790944)

# **Chapter 1**

* 1. **Introduction**

Before internet become popular like nowadays, internet only used to find information. All our affairs are troublesome, not efficient in time, and not transparent. For example, when we make a transaction in computer shop, we can be deceived by the seller because we don’t know the market price. Or we want to take a license for our business, we can be charged by illegal fees and take a long time.

But now, internet not only used to looking for information, we can see internet can take care of a lot of affairs and in many sectors. Especially with the development of Industry 4.0 with IoT technology, that internet become the backbone of that technology. The process also become efficient and more transparent. For example, Qlue is application that used for take the complaints for Jakarta residents. The complaints will be responded by relevant agencies in government and public can watch the process, whether it has been responded to or done.

This is the background of our group to used internet for administration in parking system in Faculty of Engineering

* 1. **Purpose**

Our main goal is to make parking system especially Faculty of Engineering easier. Currently, parking system using manual registration. Users have to come to parking management for registration. Registration only can be provided during working hours and users have to give the photocopy of STNK and KTM or similar identity.

With online system, users can register in anytime and anywhere so it will make parking registration easier. User only need internet connection and softcopy of STNK and KTM to make a registration. After that, parking management will get user information and verify the data.

* 1. **Target**

This program is targeted to civitas Faculty of Engineering that used motorcycle and parked it in Faculty of Engineering’s parking lot.

* 1. **Tools**
* Specification:
* Brower
* Development:
* Visual Studio Code

We will use HTML, PHP, Bootstrap, and another relevant programming language will be coded in this program. Also we use github to integrate our coding to the others.

* Xampp

Xampp is used for website simulation for front-end and back-end development.

* HARDWARE:
* PC
  1. **Risk Analysis**
* Experience problem

We must learn how to make a website, integrate it to internet, and make a relation to database. Android Studio also need to be learned to make this project.

* Time management problem

In this semester we also have project from other class that need to be finished. We must being smart in organizing our time to do this project because there are many lab works, homeworks from other class and many lessons from other class we must learn. Beside college activity, each of us follows and take part in organization.

* Collaboration problem

Because the hectic schedule, it is difficult to meet face to face to see the progress, to anticipate that we use Microsoft team for collaborative tools

**Chapter 2**

1. 1. **Project Integration Management**

Project integration management is to control all of project management through every step and process. All activities are controlled to achieve completion. And to make sure this project completed on time.

* 1. **Project Scope Management**

Scope management is needed to control this project objectives will complete. Through scope management project could be specified and structured. This project scope is to make registration become easier. It don’t need physical devices. This project mostly use software to develop the application.

* 1. **Project Time Management**

Time management is important thing we need to concern. It will consume much time to develop this project. We need to arrange our time, by planning time that we use to complete this project. The time of completion to this project

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Duty | Duration | Start | Finish | Leader |
| Planning | 3 days | 8/10/2018 | 11/10/2018 | Fadel |
| Survey | 7 Days | 12/10/2018 | 19/10/2018 | Wito |
| Design UI | 4 Days | 20/10/2018 | 24/10/2018 | Bassam |
| Develop and Design | 10 Days | 25/10/2018 | 4/11/2018 | Wito |
| Testing and Implementation | 10 Days | 5/11/2018 | 15/11/2018 | Fadel |
| Deployment and Maintenance | 14 days | 16/11/2018 | 30/11/2018 | Bassam |

* 1. **Project Cost Management**

Project Cost Management is important to allocating the budget. The cost used for this project is to rent the server and domain.

* 1. **Project Quality Management**

The user is very important to us. The quality of this project is needed to give best user experience. The ease of use of this project is developed effectively.

* 1. **Project Human Resources Management**

Project Human Resource Management involves planning project roles, responsibilities, required skills, reporting relationships, and staffing.

This table below is roles that we planned to work on this project. Every person have their own responsibilities. But it don’t make a barrier to separate us if some needed help.

|  |  |  |
| --- | --- | --- |
| Name | Role | Description |
| M Luthfi Mufadel | Project Leader, Programmer, Back-end Engineer | * Lead and guide the team * Develop program * Develop back-end * Create documentation |
| M Wito Malik | Programmer, UI Designer, Front-end Engineer, | * Develop program * Design interface * Develop front-end * Create documentation |
| J Firdaus Bassam | Programmer, Back-end Engineer, UX Designer | * Develop program * Develop back-end * Design UX * Create documentation |

* 1. **Project Communication Management**
     1. **Information Distribution**

In managing our communication in this project, we thought that we won’t meet any difficulties, considering the communication will be easy to do because we often meet at our university and the fact that our house is not that far away from each other. For other times when we can’t meet each other, we will use Line application to do online sharing and discussing our project.

* + 1. **Information Distribution**

The information of this program will be shared in Microsoft Team and Github.

* 1. **Project Risk Management**

Project risk management is knowledge about identifying, analyzing, and handling risks through the process of the project with the main purpose of meeting project objectives. Risk management has a positive impact to maintain our progress in order to finish the project.

* + 1. **Risk Identification**

The risks of this project are focused to the weakness and error-features that may be happen next. Five categories of risk are identified:

• Programming (probability : medium)

In this project we need to understand about web programming (frontend and backend). Some of us have understood the basics of the web program, so that doesn’t really matter. The big challenge for us is to make login system more simple like using email verification.

• Design (probability : low)

The most important part in the design of our project is about efficiency, how to make the system concise and easy to understand. Design in term of appearance is not so important, bad design will not too influential.

• Time shortage (probability : medium)

We need to stick in plan with our timeline. If the project progress is delayed, it will not be completed on time or we need to adjust the schedule according to the situation.

• Personal Conditions of team members (probability : low)

Members probably meet obstacles in finishing the project, for example bad health condition because of get sick.

• Satisfaction (probability : high)

It could be happen if the users are not satisfy with the application, or some features can’t be used properly.

* + 1. **Risk Quantification**

To minimize risk, our group will take preventative measure, including maintaining timeline, learning things that have not been understood, and maintaining health condition.

* + 1. **Risk Response Development**

To minimize risk, our group will take preventative measure, including maintaining timeline, learning things that have not been understood, and maintaining health condition.

* + 1. **Risk Respond**

For each points of the risk, the response that we can do is planned like this:

• For the first point, it can be overcome by the division of labor according to interests and abilities so that it will facilitate the learning process

• For the second, we should consult the matter with the people who are good with designing system

• For the third, we need to manage time well, always pay attention to the work schedule and ask other member if there are some problems to do the task so we can help each other.

• For the fourth, just like the third point we need to tell other member if there are some obstacles or bad situation.

• And for the last, we will prevent it by holding a trial period when the program has finished to see if there are bugs or dissatisfaction.

* 1. **Project Procurement Management**

This Procurement Management Plan sets the procurement framework for our project. It will serve as a guide for managing the procurements throughout the life of the project and will be updated as acquisition needs change. This plan identifies and defines the items to be procured, and decision criteria. The importance of coordinating procurement activities, and metrics in measuring procurement activities is also included.

**2.9.1 Project Planning**

The Project Leader will provide oversight and management in concert with appropriate agency procurement and management staff for all procurement activities under our project. The Project Leader will work with the project team to identify all items to be procured for the successful completion of the project.

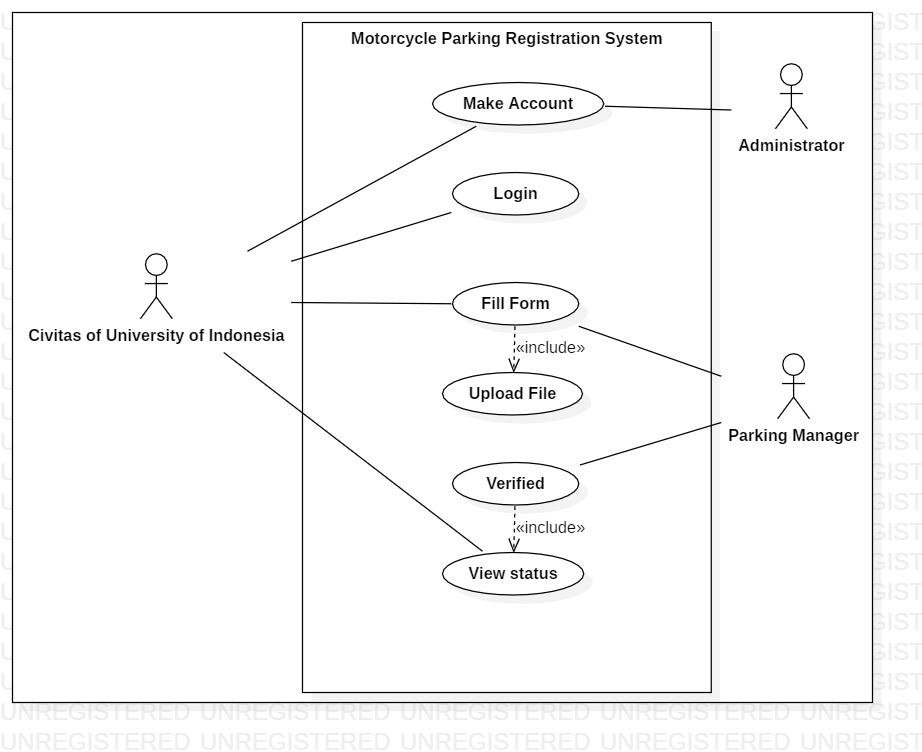
# **CHAPTER 3 DESIGN DIAGRAMS**

1. **Design Diagrams**

This chapter shows UML Diagram. UML used to explain the system by 4 views, which are requirements, logical, dynamic, and physical views, and is controlled by Object Management Group (OMG).

* 1. **Use-Case Diagram**

Use case diagrams are usually referred to as behavior diagrams used to describe a set of actions (use cases) that some system or systems (subject) should or can perform in collaboration with one or more external users of the system (actors).



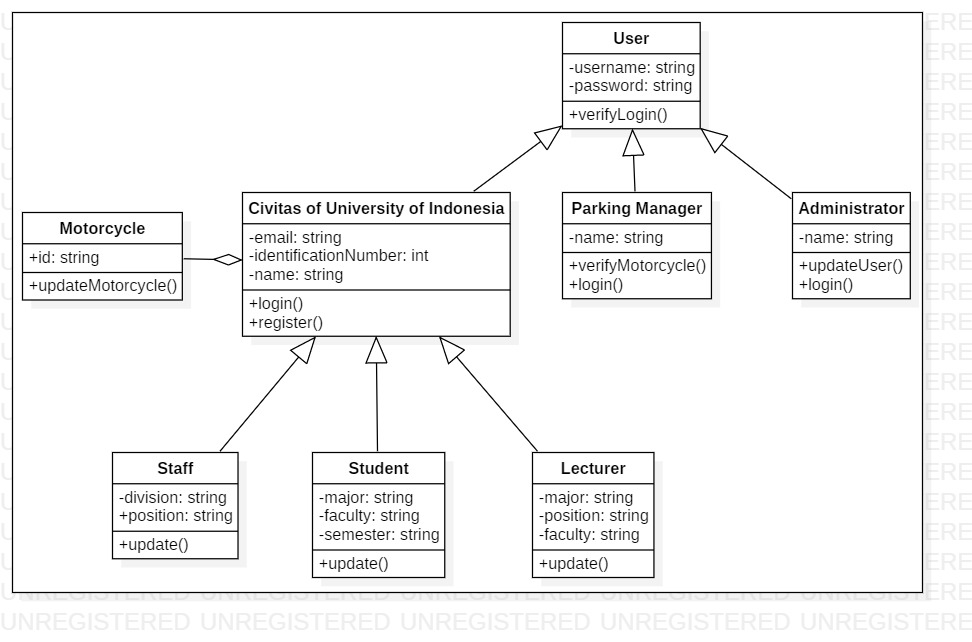
In this program, there is three actor, which is Civitas of Universitas Indonesia, Administrator, and Parking manager.. The civitas of Universitas Indonrsia has a set of actions they can do in this Website, which are to make account, login, fill form and upload file, and view status.

* 1. **Class Diagram**

Class diagram is a static diagram. It represents the static view of an application. Class diagram is not only used for visualizing, describing, and documenting different aspects of a system but also for constructing executable code of the software application.

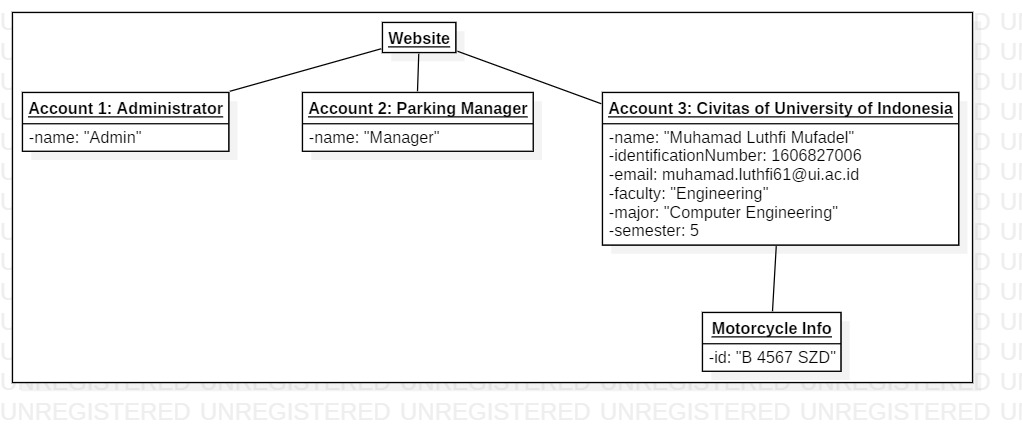
Class diagram describes the attributes and operations of a class and also the constraints imposed on the system. The class diagrams are widely used in the modeling of objectoriented systems because they are the only UML diagrams, which can be mapped directly with object-oriented languages.

Class diagram shows a collection of classes, interfaces, associations, collaborations, and constraints. It is also known as a structural diagram. This is class diagram for motorcycle parking registration system.

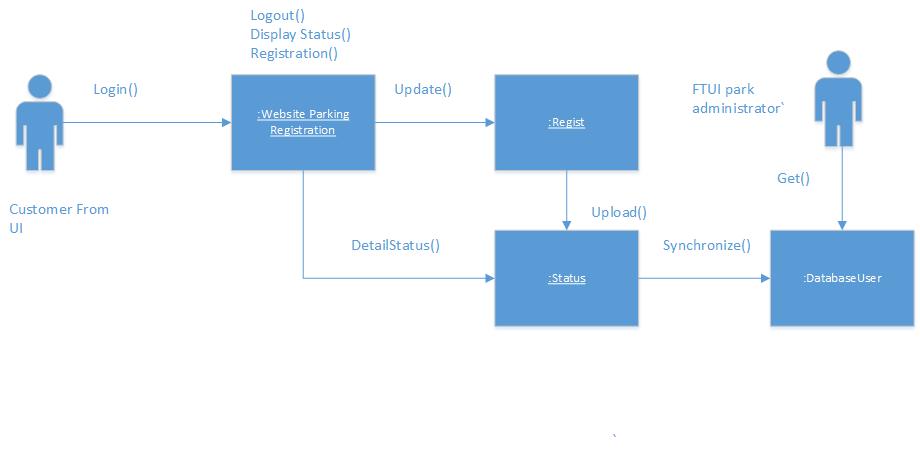


There are 8 class in our class diagram, which are motorcycle, staff, student, lecturer, civitas of Uniersitas Indonesia, parking manager, administrator, and user.

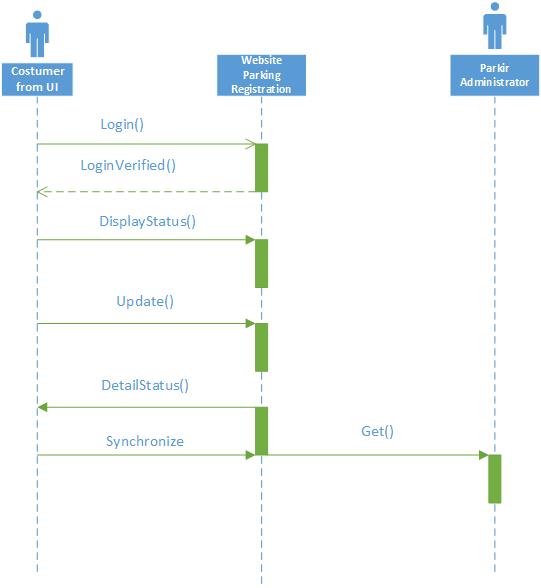
3. 3. **Object Diagram**



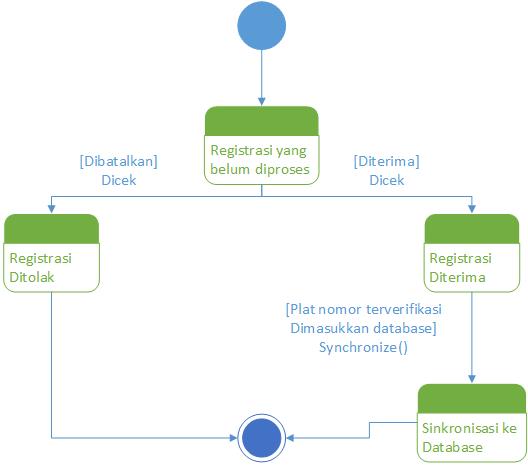
* 1. **Collaboration Diagram**



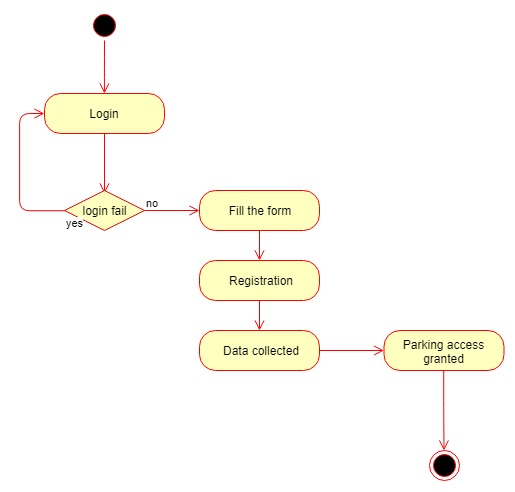
* 1. **Sequence Diagram**



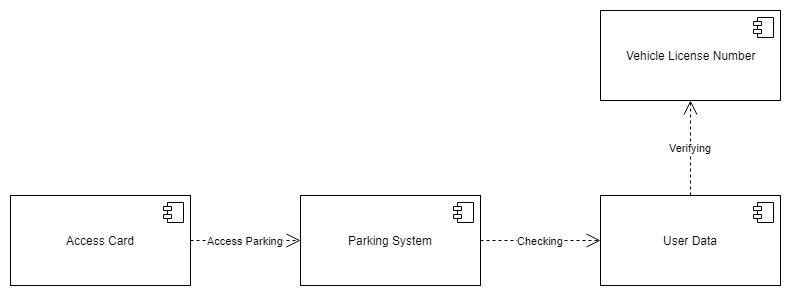
* 1. **State Diagram**



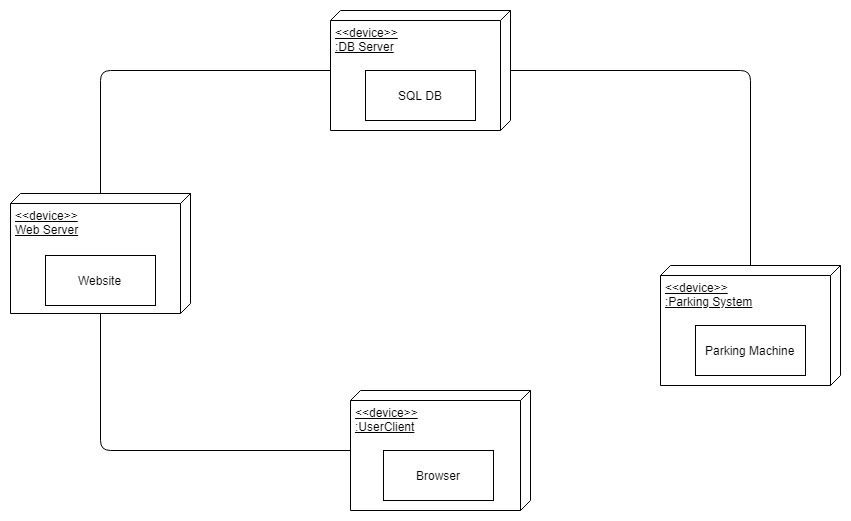
* 1. **Activity Diagram**



* 1. **Component Diagram**



3. 9. **Deployment Diagram**



# **CHAPTER 4 IMPLEMENTATION**

We use Laravel Framework in php language. We choose this framework because this is powerful and have a lot of documentations in internet. To implement this program to the internet there are few steps to do, so we include the source code and instruction for hosting implementation.

4. 1. **Source Code**

HomeController.php

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

class HomeController extends Controller

{

/\*\*

\* Create a new controller instance.

\*

\* @return void

\*/

public function \_\_construct()

{

$this->middleware('auth');

}

/\*\*

\* Show the application dashboard.

\*

\* @return \Illuminate\Http\Response

\*/

public function index()

{

return view('home');

}

}

ParkingController.php

<?php

namespace App\Http\Controllers;

use App\parking;

use Illuminate\Http\Request;

class ParkingController extends Controller

{

/\*\*

\* Display a listing of the resource.

\*

\* @return \Illuminate\Http\Response

\*/

public function index()

{

$parkings = parking::all()->toArray();

return view('parking.index', compact('parkings'));

}

/\*\*

\* Show the form for creating a new resource.

\*

\* @return \Illuminate\Http\Response

\*/

public function create()

{

return view('parking.create');

}

/\*\*

\* Store a newly created resource in storage.

\*

\* @param \Illuminate\Http\Request $request

\* @return \Illuminate\Http\Response

\*/

public function store(Request $request)

{

$this->validate($request, [

'nama\_lengkap' => 'required',

'npm' => 'required',

'plat\_nomor' => 'required'

]);

$parking = new parking([

'nama\_lengkap' => $request->get('nama\_lengkap'),

'npm' => $request->get('npm'),

'plat\_nomor' => $request->get('plat\_nomor')

]);

$parking->save();

return redirect()->route('parking.index')->with('success', 'Data Added');

}

/\*\*

\* Display the specified resource.

\*

\* @param \App\parking $parking

\* @return \Illuminate\Http\Response

\*/

public function show(parking $parking)

{

//

}

/\*\*

\* Show the form for editing the specified resource.

\*

\* @param \App\parking $parking

\* @return \Illuminate\Http\Response

\*/

public function edit($id)

{

$parking = parking::find($id);

return view('parking.edit', compact('parking', 'id'));

}

/\*\*

\* Update the specified resource in storage.

\*

\* @param \Illuminate\Http\Request $request

\* @param \App\parking $parking

\* @return \Illuminate\Http\Response

\*/

public function update(Request $request, $id)

{

$this->validate($request, [

'nama\_lengkap' => 'required',

'npm' => 'required',

'plat\_nomor' => 'required'

]);

$parking = parking::find($id);

$parking->nama\_lengkap = $request->get('nama\_lengkap');

$parking->npm = $request->get('npm');

$parking->plat\_nomor = $request->get('plat\_nomor');

$parking->save();

return redirect()->route('parking.index')->with('success', 'Data Updated');

}

/\*\*

\* Remove the specified resource from storage.

\*

\* @param \App\parking $parking

\* @return \Illuminate\Http\Response

\*/

public function destroy($id)

{

$parking = parking::find($id);

$parking->delete();

return redirect()->route('parking.index')->with('success', 'Data Deleted');

}

}

Parking.php (Model)

<?php

namespace App;

use Illuminate\Database\Eloquent\Model;

class parking extends Model

{

protected $fillable = ['nama\_lengkap', 'npm', 'plat\_nomor'];

}

2014\_10\_12\_000000\_create\_users\_table.php (migrations)

<?php

use Illuminate\Support\Facades\Schema;

use Illuminate\Database\Schema\Blueprint;

use Illuminate\Database\Migrations\Migration;

class CreateUsersTable extends Migration

{

/\*\*

\* Run the migrations.

\*

\* @return void

\*/

public function up()

{

Schema::create('users', function (Blueprint $table) {

$table->increments('id');

$table->string('name');

$table->string('email')->unique();

$table->timestamp('email\_verified\_at')->nullable();

$table->string('password');

$table->rememberToken();

$table->timestamps();

});

}

/\*\*

\* Reverse the migrations.

\*

\* @return void

\*/

public function down()

{

Schema::dropIfExists('users');

}

}

2014\_10\_12\_100000\_create\_password\_resets\_table.php (migrations)

<?php

use Illuminate\Support\Facades\Schema;

use Illuminate\Database\Schema\Blueprint;

use Illuminate\Database\Migrations\Migration;

class CreatePasswordResetsTable extends Migration

{

/\*\*

\* Run the migrations.

\*

\* @return void

\*/

public function up()

{

Schema::create('password\_resets', function (Blueprint $table) {

$table->string('email')->index();

$table->string('token');

$table->timestamp('created\_at')->nullable();

});

}

/\*\*

\* Reverse the migrations.

\*

\* @return void

\*/

public function down()

{

Schema::dropIfExists('password\_resets');

}

}

2018\_11\_28\_123917\_create\_parkings\_table.php (migrations)

<?php

use Illuminate\Support\Facades\Schema;

use Illuminate\Database\Schema\Blueprint;

use Illuminate\Database\Migrations\Migration;

class CreateParkingsTable extends Migration

{

/\*\*

\* Run the migrations.

\*

\* @return void

\*/

public function up()

{

Schema::create('parkings', function (Blueprint $table) {

$table->increments('id');

$table->text('nama\_lengkap');

$table->integer('npm');

$table->text('plat\_nomor');

$table->timestamps();

});

}

/\*\*

\* Reverse the migrations.

\*

\* @return void

\*/

public function down()

{

Schema::dropIfExists('parkings');

}

}

email.blade.php (view)

@extends('layouts.app')

@section('content')

<div class="container">

<div class="row justify-content-center">

<div class="col-md-8">

<div class="card">

<div class="card-header">{{ \_\_('Reset Password') }}</div>

<div class="card-body">

@if (session('status'))

<div class="alert alert-success" role="alert">

{{ session('status') }}

</div>

@endif

<form method="POST" action="{{ route('password.email') }}">

@csrf

<div class="form-group row">

<label for="email" class="col-md-4 col-form-label text-md-right">{{ \_\_('E-Mail Address') }}</label>

<div class="col-md-6">

<input id="email" type="email" class="form-control{{ $errors->has('email') ? ' is-invalid' : '' }}" name="email" value="{{ old('email') }}" required>

@if ($errors->has('email'))

<span class="invalid-feedback" role="alert">

<strong>{{ $errors->first('email') }}</strong>

</span>

@endif

</div>

</div>

<div class="form-group row mb-0">

<div class="col-md-6 offset-md-4">

<button type="submit" class="btn btn-primary">

{{ \_\_('Send Password Reset Link') }}

</button>

</div>

</div>

</form>

</div>

</div>

</div>

</div>

</div>

@endsection

reset.blade.php (view)

@extends('layouts.app')

@section('content')

<div class="container">

<div class="row justify-content-center">

<div class="col-md-8">

<div class="card">

<div class="card-header">{{ \_\_('Reset Password') }}</div>

<div class="card-body">

<form method="POST" action="{{ route('password.update') }}">

@csrf

<input type="hidden" name="token" value="{{ $token }}">

<div class="form-group row">

<label for="email" class="col-md-4 col-form-label text-md-right">{{ \_\_('E-Mail Address') }}</label>

<div class="col-md-6">

<input id="email" type="email" class="form-control{{ $errors->has('email') ? ' is-invalid' : '' }}" name="email" value="{{ $email ?? old('email') }}" required autofocus>

@if ($errors->has('email'))

<span class="invalid-feedback" role="alert">

<strong>{{ $errors->first('email') }}</strong>

</span>

@endif

</div>

</div>

<div class="form-group row">

<label for="password" class="col-md-4 col-form-label text-md-right">{{ \_\_('Password') }}</label>

<div class="col-md-6">

<input id="password" type="password" class="form-control{{ $errors->has('password') ? ' is-invalid' : '' }}" name="password" required>

@if ($errors->has('password'))

<span class="invalid-feedback" role="alert">

<strong>{{ $errors->first('password') }}</strong>

</span>

@endif

</div>

</div>

<div class="form-group row">

<label for="password-confirm" class="col-md-4 col-form-label text-md-right">{{ \_\_('Confirm Password') }}</label>

<div class="col-md-6">

<input id="password-confirm" type="password" class="form-control" name="password\_confirmation" required>

</div>

</div>

<div class="form-group row mb-0">

<div class="col-md-6 offset-md-4">

<button type="submit" class="btn btn-primary">

{{ \_\_('Reset Password') }}

</button>

</div>

</div>

</form>

</div>

</div>

</div>

</div>

</div>

@endsection

login.blade.php (view)

@extends('layouts.app')

@section('content')

<div class="container">

<div class="row justify-content-center">

<div class="col-md-8">

<div class="card">

<div class="card-header">{{ \_\_('Login') }}</div>

<div class="card-body">

<form method="POST" action="{{ route('login') }}">

@csrf

<div class="form-group row">

<label for="email" class="col-sm-4 col-form-label text-md-right">{{ \_\_('E-Mail Address') }}</label>

<div class="col-md-6">

<input id="email" type="email" class="form-control{{ $errors->has('email') ? ' is-invalid' : '' }}" name="email" value="{{ old('email') }}" required autofocus>

@if ($errors->has('email'))

<span class="invalid-feedback" role="alert">

<strong>{{ $errors->first('email') }}</strong>

</span>

@endif

</div>

</div>

<div class="form-group row">

<label for="password" class="col-md-4 col-form-label text-md-right">{{ \_\_('Password') }}</label>

<div class="col-md-6">

<input id="password" type="password" class="form-control{{ $errors->has('password') ? ' is-invalid' : '' }}" name="password" required>

@if ($errors->has('password'))

<span class="invalid-feedback" role="alert">

<strong>{{ $errors->first('password') }}</strong>

</span>

@endif

</div>

</div>

<div class="form-group row">

<div class="col-md-6 offset-md-4">

<div class="form-check">

<input class="form-check-input" type="checkbox" name="remember" id="remember" {{ old('remember') ? 'checked' : '' }}>

<label class="form-check-label" for="remember">

{{ \_\_('Remember Me') }}

</label>

</div>

</div>

</div>

<div class="form-group row mb-0">

<div class="col-md-8 offset-md-4">

<button type="submit" class="btn btn-primary">

{{ \_\_('Login') }}

</button>

@if (Route::has('password.request'))

<a class="btn btn-link" href="{{ route('password.request') }}">

{{ \_\_('Forgot Your Password?') }}

</a>

@endif

</div>

</div>

</form>

</div>

</div>

</div>

</div>

</div>

@endsection

register.blade.php (view)

@extends('layouts.app')

@section('content')

<div class="container">

<div class="row justify-content-center">

<div class="col-md-8">

<div class="card">

<div class="card-header">{{ \_\_('Register') }}</div>

<div class="card-body">

<form method="POST" action="{{ route('register') }}">

@csrf

<div class="form-group row">

<label for="name" class="col-md-4 col-form-label text-md-right">{{ \_\_('Name') }}</label>

<div class="col-md-6">

<input id="name" type="text" class="form-control{{ $errors->has('name') ? ' is-invalid' : '' }}" name="name" value="{{ old('name') }}" required autofocus>

@if ($errors->has('name'))

<span class="invalid-feedback" role="alert">

<strong>{{ $errors->first('name') }}</strong>

</span>

@endif

</div>

</div>

<div class="form-group row">

<label for="email" class="col-md-4 col-form-label text-md-right">{{ \_\_('E-Mail Address') }}</label>

<div class="col-md-6">

<input id="email" type="email" class="form-control{{ $errors->has('email') ? ' is-invalid' : '' }}" name="email" value="{{ old('email') }}" required>

@if ($errors->has('email'))

<span class="invalid-feedback" role="alert">

<strong>{{ $errors->first('email') }}</strong>

</span>

@endif

</div>

</div>

<div class="form-group row">

<label for="password" class="col-md-4 col-form-label text-md-right">{{ \_\_('Password') }}</label>

<div class="col-md-6">

<input id="password" type="password" class="form-control{{ $errors->has('password') ? ' is-invalid' : '' }}" name="password" required>

@if ($errors->has('password'))

<span class="invalid-feedback" role="alert">

<strong>{{ $errors->first('password') }}</strong>

</span>

@endif

</div>

</div>

<div class="form-group row">

<label for="password-confirm" class="col-md-4 col-form-label text-md-right">{{ \_\_('Confirm Password') }}</label>

<div class="col-md-6">

<input id="password-confirm" type="password" class="form-control" name="password\_confirmation" required>

</div>

</div>

<div class="form-group row mb-0">

<div class="col-md-6 offset-md-4">

<button type="submit" class="btn btn-primary">

{{ \_\_('Register') }}

</button>

</div>

</div>

</form>

</div>

</div>

</div>

</div>

</div>

@endsection

verify.blade.php (view)

@extends('layouts.app')

@section('content')

<div class="container">

<div class="row justify-content-center">

<div class="col-md-8">

<div class="card">

<div class="card-header">{{ \_\_('Verify Your Email Address') }}</div>

<div class="card-body">

@if (session('resent'))

<div class="alert alert-success" role="alert">

{{ \_\_('A fresh verification link has been sent to your email address.') }}

</div>

@endif

{{ \_\_('Before proceeding, please check your email for a verification link.') }}

{{ \_\_('If you did not receive the email') }}, <a href="{{ route('verification.resend') }}">{{ \_\_('click here to request another') }}</a>.

</div>

</div>

</div>

</div>

</div>

@endsection

app.blade.php (view)

<!DOCTYPE html>

<html lang="{{ str\_replace('\_', '-', app()->getLocale()) }}">

<head>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1">

<!-- CSRF Token -->

<meta name="csrf-token" content="{{ csrf\_token() }}">

<title>{{ config('app.name', 'Laravel') }}</title>

<!-- Scripts -->

<script src="{{ asset('js/app.js') }}" defer></script>

<!-- Fonts -->

<link rel="dns-prefetch" href="//fonts.gstatic.com">

<link href="https://fonts.googleapis.com/css?family=Nunito" rel="stylesheet" type="text/css">

<!-- Styles -->

<link href="{{ asset('css/app.css') }}" rel="stylesheet">

</head>

<body>

<div id="app">

<nav class="navbar navbar-expand-md navbar-dark navbar-laravel">

<div class="container">

@if (Route::has('login'))

@auth

<a class="navbar-brand" href="{{ url('/home') }}">

{{ config('app.name', 'Laravel') }}

</a>

@else

<a class="navbar-brand" href="{{ url('/') }}">

{{ config('app.name', 'Laravel') }}

</a>

@endauth

@endif

<button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="{{ \_\_('Toggle navigation') }}">

<span class="navbar-toggler-icon"></span>

</button>

<div class="collapse navbar-collapse" id="navbarSupportedContent">

<!-- Left Side Of Navbar -->

<ul class="navbar-nav mr-auto">

</ul>

<!-- Right Side Of Navbar -->

<ul class="navbar-nav ml-auto">

<!-- Authentication Links -->

@guest

<li class="nav-item">

<a class="nav-link" href="{{ route('login') }}">{{ \_\_('Login') }}</a>

</li>

<li class="nav-item">

@if (Route::has('register'))

<a class="nav-link" href="{{ route('register') }}">{{ \_\_('Register') }}</a>

@endif

</li>

@else

<li class="nav-item dropdown">

<a id="navbarDropdown" class="nav-link dropdown-toggle" href="#" role="button" data-toggle="dropdown" aria-haspopup="true" aria-expanded="false" v-pre>

{{ Auth::user()->name }} <span class="caret"></span>

</a>

<div class="dropdown-menu dropdown-menu-right" aria-labelledby="navbarDropdown">

<a class="dropdown-item" href="{{ url('/information') }}">Information</a>

<a class="dropdown-item" href="{{ route('logout') }}"

onclick="event.preventDefault();

document.getElementById('logout-form').submit();">

{{ \_\_('Logout') }}

</a>

<form id="logout-form" action="{{ route('logout') }}" method="POST" style="display: none;">

@csrf

</form>

</div>

</li>

@endguest

</ul>

</div>

</div>

</nav>

<main class="py-4">

@yield('content')

</main>

</div>

</body>

</html>

Create.blade.php (view)

@extends('master')

@section('content')

<div class="row">

<div class="col-md-12">

<br />

<h3 aling="center">Fill the form</h3>

<br />

@if(count($errors) > 0)

<div class="alert alert-danger">

<ul>

@foreach($errors->all() as $error)

<li>{{$error}}</li>

@endforeach

</ul>

</div>

@endif

@if(\Session::has('success'))

<div class="alert alert-success">

<p>{{ \Session::get('success') }}</p>

</div>

@endif

<form method="post" action="{{url('parking')}}">

{{csrf\_field()}}

<div class="form-group">

<input type="text" name="nama\_lengkap" class="form-control" placeholder="Masukkan Nama Lengkap" />

</div>

<div class="form-group">

<input type="number" name="npm" class="form-control" placeholder="Masukkan NPM" />

</div>

<div class="form-group">

<input type="text" name="plat\_nomor" class="form-control" placeholder="Masukkan Plat Nomor" />

</div>

<div class="form-group">

<input type="submit" class="btn btn-primary" />

<input type="button" class="btn btn-primary" onclick="location.href='/rpl/public/home';" value='Home'/>

<input type="button" class="btn btn-primary" onclick="location.href='/rpl/public/parking';" value='Check'/>

</div>

</form>

</div>

</div>

@endsection

Edit.blade.php (view)

@extends('master')

@section('content')

<div class="row">

<div class="col-md-12">

<br />

<h3>Edit Record</h3>

<br />

@if(count($errors) > 0)

<div class="alert alert-danger">

<ul>

@foreach($errors->all() as $error)

<li>{{$error}}</li>

@endforeach

</ul>

@endif

<form method="post" action="{{action('ParkingController@update', $id)}}">

{{csrf\_field()}}

<input type="hidden" name="\_method" value="PATCH" />

<div class="form-group">

<input type="text" name="nama\_lengkap" class="form-control" value="{{$parking->nama\_lengkap}}" placeholder="Masukkan Nama Lengkap" />

</div>

<div class="form-group">

<input type="number" name="npm" class="form-control" value="{{$parking->npm}}" placeholder="Masukkan NPM" />

</div>

<div class="form-group">

<input type="text" name="plat\_nomor" class="form-control" value="{{$parking->plat\_nomor}}" placeholder="Masukkan Plat Nomor" />

</div>

<div class="form-group">

<input type="submit" class="btn btn-primary" value="Edit" />

<input type="button" class="btn btn-primary" onclick="location.href='/rpl/public/parking';" value='Back'/>

</div>

</form>

</div>

</div>

@endsection

Index.blade.php (view)

@extends('master')

@section('content')

<div class="row">

<div class="col-md-12">

<br />

<h3 align="center">Parking Data</h3>

<br />

@if($message = Session::get('success'))

<div class="alert alert-success">

<p>{{$message}}</p>

</div>

@endif

<div align="right">

<a href="{{route('parking.create')}}" class="btn btn-primary">Add</a>

<br />

<br />

</div>

<table class="table table-bordered table-striped">

<tr>

<th>Nama Lengkap</th>

<th>NPM</th>

<th>Plat Nomor</th>

<th>Edit</th>

<th>Delete</th>

</tr>

@foreach($parkings as $row)

<tr>

<td>{{$row['nama\_lengkap']}}</td>

<td>{{$row['npm']}}</td>

<td>{{$row['plat\_nomor']}}</td>

<td><a href="{{action('ParkingController@edit', $row['id'])}}" class="btn btn-warning">Edit</a></td>

<td>

<form method="post" class="delete\_form" action="{{action('ParkingController@destroy', $row['id'])}}">

{{csrf\_field()}}

<input type="hidden" name="\_method" value="DELETE" />

<button type="submit" class="btn btn-danger">Delete</button>

</form>

</td>

</tr>

@endforeach

</table>

<a href="/rpl/public/home" class="btn btn-primary">Home</a>

</div>

</div>

<script>

$(document).ready(function(){

$('.delete\_form').on('submit', function(){

if(confirm("Are you sure you want to delete it?"))

{

return true;

}

else

{

return false;

}

});

});

</script>

@endsection

Home.blade.php (view)

@extends('layouts.app')

@section('content')

<div class="container">

<div class="row justify-content-center">

<div class="col-md-8">

<div class="card">

<div class="card-header">Dashboard</div>

<div class="card-body">

@if (session('status'))

<div class="alert alert-success" role="alert">

{{ session('status') }}

</div>

@endif

You are logged in!

<br><br>

<a href="/rpl/public/parking" class="btn btn-primary">Parking Registration</a>

</div>

</div>

</div>

</div>

</div>

@endsection

Information.blade.php (view)

@extends('layouts.app')

@section('content')

<div class="container">

<div class="row justify-content-center">

<div class="col-md-8">

<div class="jumbotron">

<h1 class="display-4">Pagi Teknik!</h1>

<p class="lead">This is a simple hero unit, a simple jumbotron-style component for calling extra attention to featured content or information.</p>

<hr class="my-4">

<p>It uses utility classes for typography and spacing to space content out within the larger container.</p>

@if (Route::has('login'))

<div>

@auth

@else

<a class="btn btn-primary btn-lg" role="button" href="{{ route('register') }}" aria-pressed="true">Register</a>

<a href="{{ route('login') }}" class="btn btn-secondary btn-lg active" role="button" aria-pressed="true">Login</a>

@endauth

</div>

@endif

</div>

</div>

</div>

</div>

@endsection

Master.blade.php (view)

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>Motorcycle Parking Registration System</title>

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>

<link href="https://cdnjs.cloudflare.com/ajax/libs/twitter-bootstrap/4.0.0-alpha/css/bootstrap.css" rel="stylesheet">

</head>

<body>

<div class="container">

@yield('content')

</div>

</body>

</html>

Welcome.blade.php (view)

<!doctype html>

<html lang="{{ str\_replace('\_', '-', app()->getLocale()) }}">

<head>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>FTUI Parking Registration System</title>

<!-- Fonts -->

<link href="https://fonts.googleapis.com/css?family=Nunito:200,600" rel="stylesheet" type="text/css">

<!-- Styles -->

<link href="{{ asset('css/app.css') }}" rel="stylesheet">

<link href="{{ asset('css/cover.css') }}" rel="stylesheet">

</head>

<body class="text-center">

<div class="cover-container d-flex w-100 h-100 p-3 mx-auto flex-column">

<header class="masthead mb-auto">

<div class="inner">

<nav class="nav nav-masthead justify-content-center">

@if (Route::has('login'))

@auth

@else

<a class="nav-link" href="{{ route('login') }}">Login</a>

@if (Route::has('register'))

<a class="nav-link" href="{{ route('register') }}">Register</a>

@endif

@endauth

@endif

</nav>

</div>

</header>

<main role="main" class="inner cover">

<h1 class="cover-heading"><b>FTUI</b> Parking Registration System</h1>

<p class="lead">Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.</p>

@if (Route::has('login'))

@auth

<p class="lead">

<a class="btn btn-lg btn-secondary" href="{{ url('/home') }}">Dashboard</a>

</p>

@else

<p class="lead">

<a class="btn btn-lg btn-secondary" href="{{ url('/information') }}">Information</a>

</p>

@endauth

@endif

</main>

<footer class="mastfoot mt-auto">

<div class="inner">

<p>FTUI Parking Registration System</p>

</div>

</footer>

</div>

</body>

</html>

Web.php (routes)

<?php

/\*

|--------------------------------------------------------------------------

| Web Routes

|--------------------------------------------------------------------------

|

| Here is where you can register web routes for your application. These

| routes are loaded by the RouteServiceProvider within a group which

| contains the "web" middleware group. Now create something great!

|

\*/

Route::get('/', function () {

return view('welcome');

});

Auth::routes();

Route::get('/home', 'HomeController@index')->name('home');

Route::get('/information', function () {

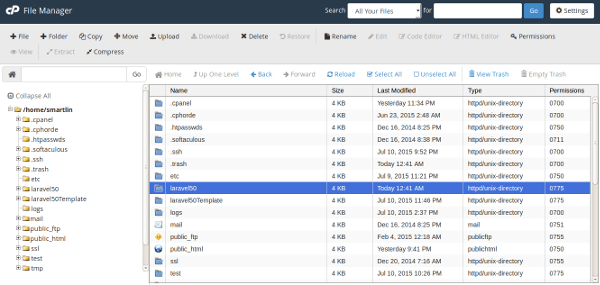
return view('information');

});

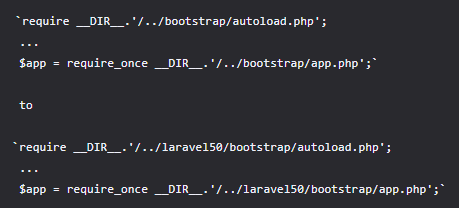
Route::resource('parking', 'ParkingController');

* 1. **Implement to Hosting Service**

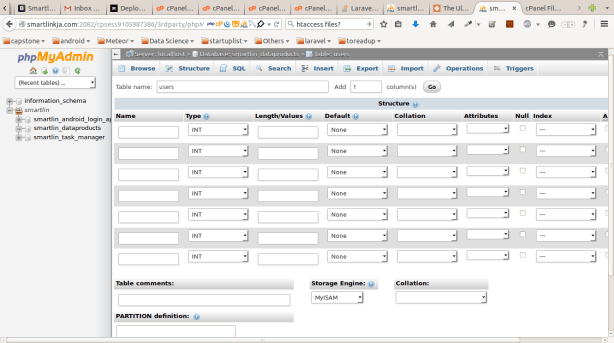
To implement this project to hosting first we need to Compress the entire project folder on your local machine. And in hosting we can upload the zip folder to the rot directory.



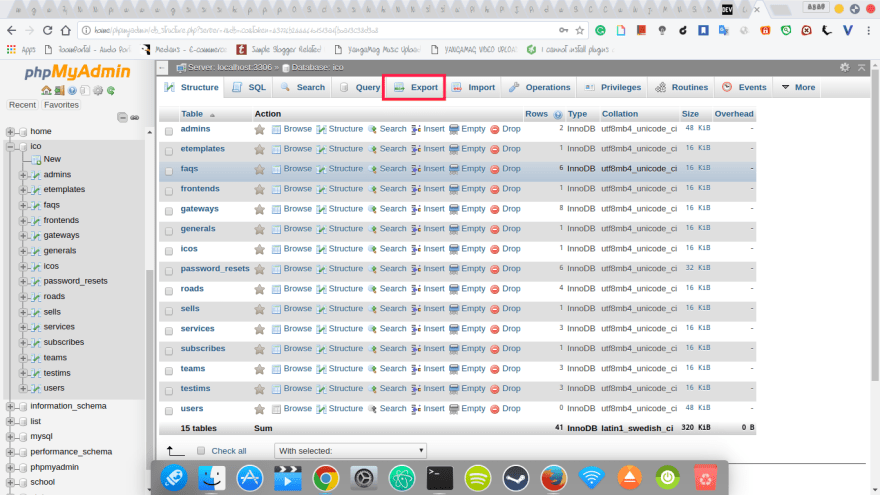
Open the extracted folder and MOVE the CONTENTS of the public folder to your cpanel’s public\_html folder. You can as well delete the empty public folder now. Navigate to the public\_html folder and locate the index.php file. Right click on it and select Code Edit from the menu. This will open up another tab showing the cpanel code editor. change the following lines (22 and 36) from. the .htaccess file should look something like this.

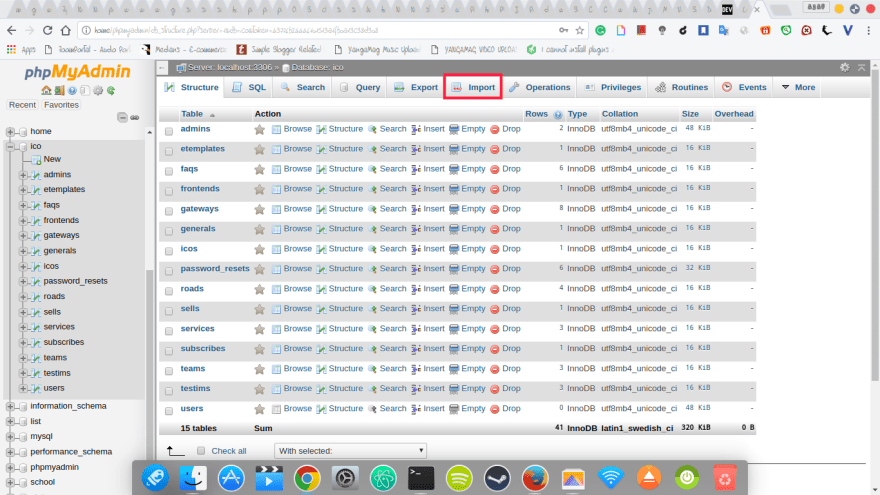


If everything went well, going to http://yourdomain.com should throw database errors. Not to worry! The next phase is migrating your databases to your shared hosting. One sweet thing about Laravel Framework is that it enables you to quickly setup defined databases with a single command php artisan migrate. Since we are using a shared hosting, we can not do this without the use of SSH. This can be done easily if you have up to 10 tables. Most cPanel comes with PHPMyAdmin and Mysql Database Wizard. Use the Mysql Database wizard to create a [Database and User] then assign the user to the database allowing all privileges. Note down the username and password you'll need that soon. Use the cpanel's PHPMyAdmin to create your tables. To do this efficiently, open up PHPMyAdmin on your local machine. For each table structure create the exact structure on your cPanel’s PHPMyAdmin.



There's another way of importing and exporting databases from a local machine. login to PHPMyAdmin from your computer locate the database your project uses, from the toolbar find the export tool.





After exporting move over to your shared hosting and do the same, but this time we are using the import tool If successfully uploaded you'll see the 'Success Message". On your cpanel file manager, navigate to \*project\* (or your laravel project directory). Go to config/database.php . Right click and select ‘Code Edit’. Locate line 55 – Your MySQL configuration Section. Change the username and password to your MySQL username and password from your cpanel. Ensure that these details are entered correctly.

# **CHAPTER 5 TESTING ANALYSIS**

* 1. **Introduction**

In this chapter is used to track the necessary information required to effectively define the testing of the website. The test plan document is created to make the website work well. This tested do by the project team. Other purpose in this chapter is to check and maintain this website if there are some advice or development that can bring more efficient and interested.

* 1. **Compatibility Testing**
     1. **Test Risk / Issues**

Motorcycle Parking Registration is web-based project to easily Universitas Indonesia civitas to register their motorcycle into FTUI park. To ensure that our program is compatible in various browser, we test our program in many browser like chrome, edge, and internet explorer.

* + 1. **Items to be Tested**

|  |  |  |  |
| --- | --- | --- | --- |
| Item to Test | Test Description | Test Date | Responsibility |
| Website | Does it run well in multiple browser | November 30, 2018 | * Jauharil Firdaus Bassam * M. Wito Malik * M. Luthfi Mufadel |

* + 1. **Test Approach**

This is important in part of test plan. The purpose is to make sure the code can run well and perfectly in many Browser. This will show development team, is there any problem, bug, or error while run this code or not. We can fix it and make sure there are no problem while run this code in the future.

* + 1. **Test Regulatory / Mandate Criteria**

To run the test, we need to install various browser and connect to xampp server and start Laravel with command “php artisan:serve” and running http://localhost:8080/cobain/public/ on all browser.

* + 1. **Test Pass / Fail Criteria**

Compatibility check make sure the game can run well in many different Web Browser also determined passed if all function of the web work well and correctly, such as web display, information, login and register function, and many else. This test is failed if the website did not run perfectly during test. If there is one function did not run properly then the code of our website must be fix soon.

* + 1. **Test Entry / Exit Criteria**

We can run the test if:

* Xampp apache and database server is running

Or

* Laravel server is running with command “php artisan:serve”

We can stop the test when:

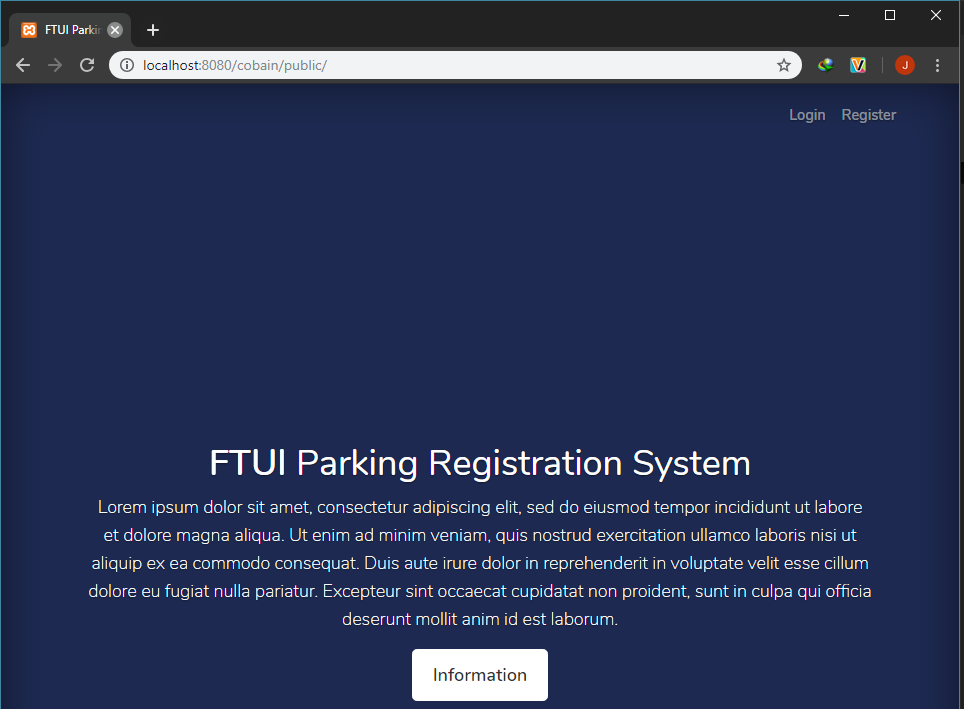
website can run well in many web browser

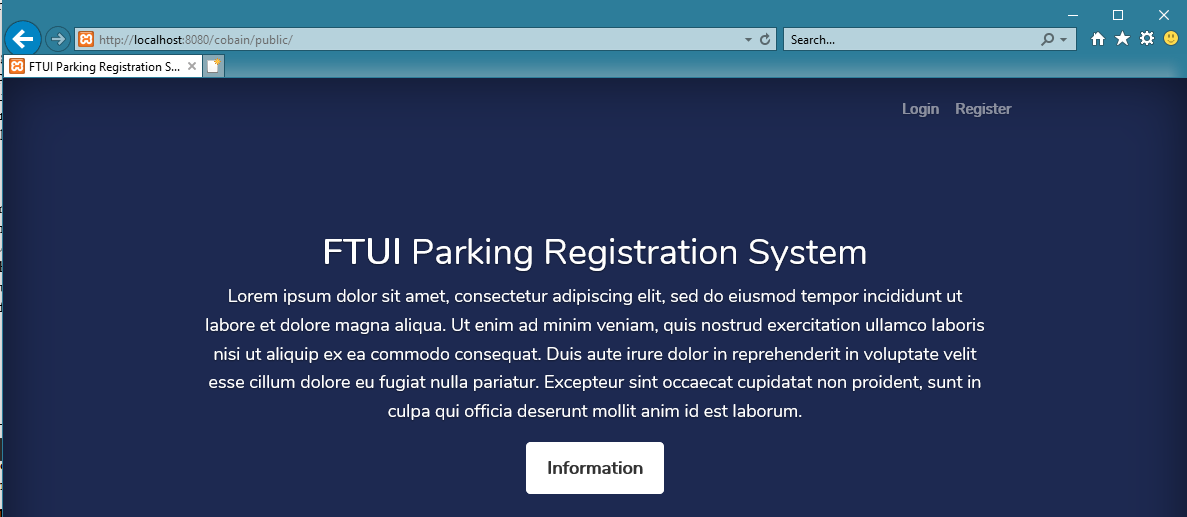
* + 1. **Test Deliverables**

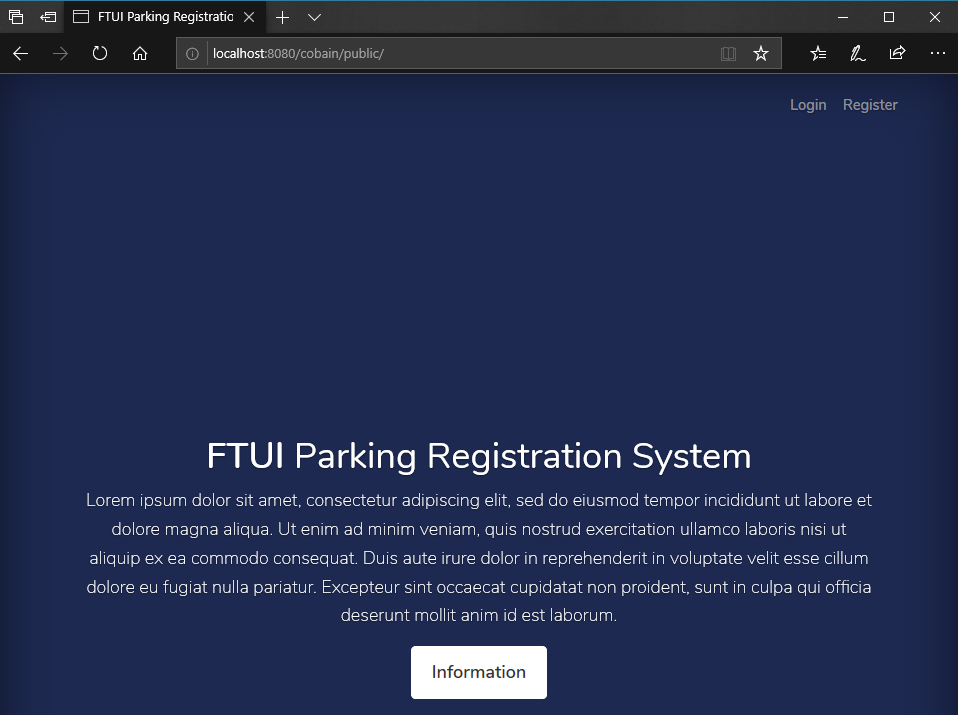
The result of this test is delivered with a documentation, including device’s specification, running-apps, screenshot and error analysis.

The result is:

|  |  |  |  |
| --- | --- | --- | --- |
| No | Device’s Name | Device’s Web Browser | Passed or Failed |
| 1 | Dell inspiron 7567 | Chrome | Passed |
| 2 | Dell inspiron 7567 | Edge | Passed |
| 3 | Dell inspiron 7567 | Internet Explorer | Passed |
| 4 |  |  |  |
| 5 |  |  |  |







* + 1. **Test Suspension**

The test will be suspended when there is any obstacles/problem such as availability of the devices that supports the test and the member’s availability. The test will be resumed if all requirement that mentioned above fulfilled.

* + 1. **Test Environmental**

This test is for supporting our program which is “FTUI Park Registration System” running in various web browser. No special training or action for run this test, but all responsible team members should learn php, Laravel framework installation, and github deeply and other general knowledge which it needed to create this website.

* 1. **Functional Testing**
     1. **Test Risks / Issues**

The problem may face with some internal algorithm or code syntax issue. Human error to determining the test failed or passed is a common factor of error in this test. To reduce some error, out team will recheck our functional code before doing test. We will test some function in our website, such as web display,information, register, and login. If there is another error or bug, we will fix it.

* + 1. **Item to be Tested**

|  |  |  |  |
| --- | --- | --- | --- |
| Item to be Tested | Test Description | Test Date | Responsibility |
| Information Page | The page showed correctly | 30 November 2018 | Jauharil Firdaus Bassam |
| Login Page | The page showed correctly | 30 November 2018 | M. Wito Malik |
| Registration Page | The page showed correctly | 30 November 2018 | M. Wito Malik |
| Registration Process | Registration process is working and user data saved in database | 30 November 2018 | Jauharil Firdaus Bassam |
| Login Process | Login process is perfectly working. If user input correct data, web page will show success notification and user , and if false web page will show failed notification | 30 November 2018 | M. Wito Malik |
| Logout Process | Logout process is working perfectly. | 30 November 2018 | M. Wito Malik |
| Parking Page | The page showed correctly | 30 November 2018 | Jauharil Firdaus Bassam |
| Create Parking Page | The page showed correctly | 30 November 2018 | M. Wito Malik |
| Edit Parking Page | The page showed correctly | 30 November 2018 | Jauharil Firdaus Bassam |
| Create Parking Process | Create parking process is working and motorcycle data saved in database | 30 November 2018 | M. Luthfi Mufadel |
| Edit Parking Process | Edit parking process is working and motorcycle data saved in database | 30 November 2018 | M. Luthfi Mufadel |
| Delete Parking Process | Delete parking process is working and motorcycle data deleted in database | 30 November 2018 | M. Luthfi Mufadel |

* + 1. **Test Approach**

The purpose is to compare the website with what the documentation of website mentioned above. Our main testing is to synchronize the register database which is obtained from register function to login function so user can login on our website . The function login,register, and motorcycle register is our main function of this website, if it run perfectly and correctly, this website can run as we hope in the start.

* + 1. **Test Regulatory / Mandate Criteria**

The minimum criteria for running in this test is that device have passed the compatibility test.

* + 1. **Test Pass / Fail Criteria**

The test succeeded if the function that mentioned above run perfectly and correctly, and there is no bug or error while running the website.

* + 1. **Test Entry / Exit Criteria**

The test starts when the websiite is being ran in the web browser, and will be stopped when all function that mentioned above work perfectly and there is no error or bug while run the website. The test will be stopped too when there is a bug or error while running the game.

* + 1. **Test Deliverables**

This test will deliver testing documentation which is how many trials, problems encountered with solution, and pass or failed.

The result is as follows:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tester Number | How many trials | Problems encountered | Solution | Passed or failed |
| 1 | 2 | 0 | - | Passed |
| 2 | 2 | 0 | - | Passed |
| 3 | 2 | 0 | - | Passed |
| 4 | 2 | 0 | - | Passed |
| 5 | 2 | 0 | - | Passed |
| 6 | 2 | 0 | - | Passed |
| 7 | 2 | 0 | - | Passed |
| 8 | 2 | 0 | - | Passed |
| 9 | 2 | 0 | - | Passed |
| 10 | 2 | 0 | - | Passed |
| 11 | 2 | 0 | - | Passed |
| 12 | 2 | 0 | - | Passed |

* + 1. **Test Suspension / Resumption Criteria**

All function testing will be suspended if the team have plenty of another important agenda to do also if there is one or more thing that can implement to our website, We will suspend the test and we will implement it first to the website.

* + 1. **Test Environmental / Staffing /Training Needs**

There is no requirement, but user should learn and know what the website function and what function use in the website like login or register.

# **CHAPTER 6 USER MANUAL AND CONTRACT**

This chapter explain about user manual and show the contract between the client with  
developer team.

## **6.1 USER MANUAL Organizational Of Manual**

1.0: General Information

This section contains general information about the our web based project “parking registration system”, including the system overview, points of contact, project references.

2.0: Getting Started

This section contains the information to get started in using parkir registration system.

### **6.1.1 General Information**

Parking registration system is website to easily any civitas FTUI to register their motorcycle to FTUI park with an online method.

### **6.1.2 Project References**

We develop the program by learning from this references:

* <https://github.com/bradtraversy/lsapp>
* <https://www.traversymedia.com/>
* https://github.com/RistekCSUI/SSO

### **6.1.3 Authorized Use Permission**

### Some of content at the website relate to register motor cycle process can be accessed by anyone and content relate to verify the registration just can be accessed by the park administrator.

## **6.1.4 Points of Contact**

## **6.1.4.1 Information**

Provide a list of the points of organizational contact (POCs) that may be needed by the  
document user for informational and troubleshooting purposes. Include type of contact, contact  
name, telephone number, and e-mail address (if applicable). Points of contact may include, but  
are not limited to, help desk POC, development/maintenance POC, and operations POC

Muhamad Luthfi Mufadel

As Project Leader, Programmer, Backend-programmer

• Email :   
• Handphone number :

### **6.1.4.2 Help Desk**

Provide help desk information including responsible developer’s phone numbers for  
emergency assistance.

1. Jauharil Firdaus Bassam – as Programmer, Programmer, Backend-programmer

* Email : [jfbassam50@gmail.com](mailto:jfbassam50@gmail.com)
* Handphone number : 08872492592

1. Muhammad Wito Malik – as Programmer, UI Designer, Frontend-programmer

* Email :
* Handphone number :

## **6.1.5 Getting Started**

### **6.1.5.1 Installation**

Parking Web Registration system designed to run in full compatibility in multiple browser like chrome, edge, etc and multiple device like pc or android.

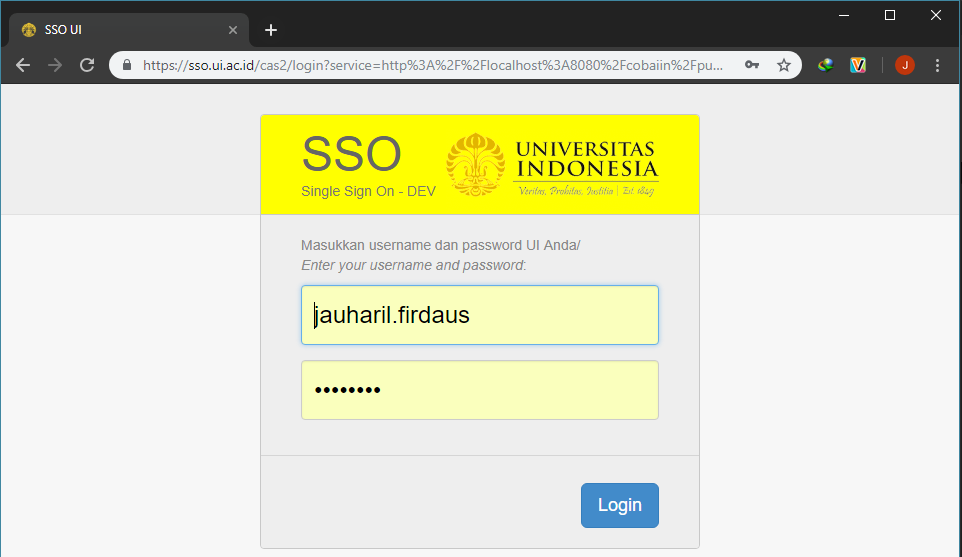
**6.1.5.2 User mode: Website homepage launch**

Connect your computer to any network because we need it to open sso ui. Now, this is our homepage (/public)



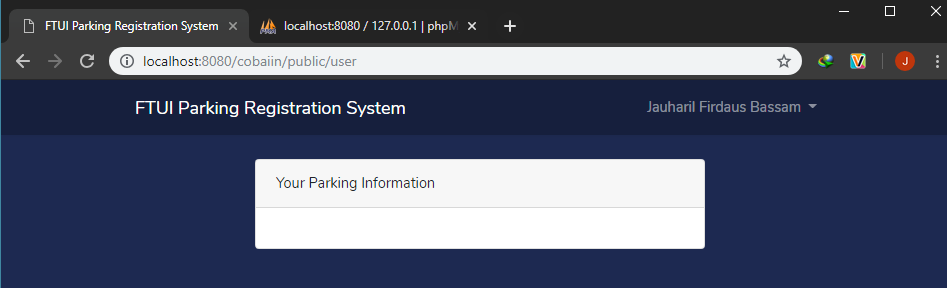
### **6.1.5.3 User Mode: Login page**

After open homepage, you can login with your sso account with click “login” button.

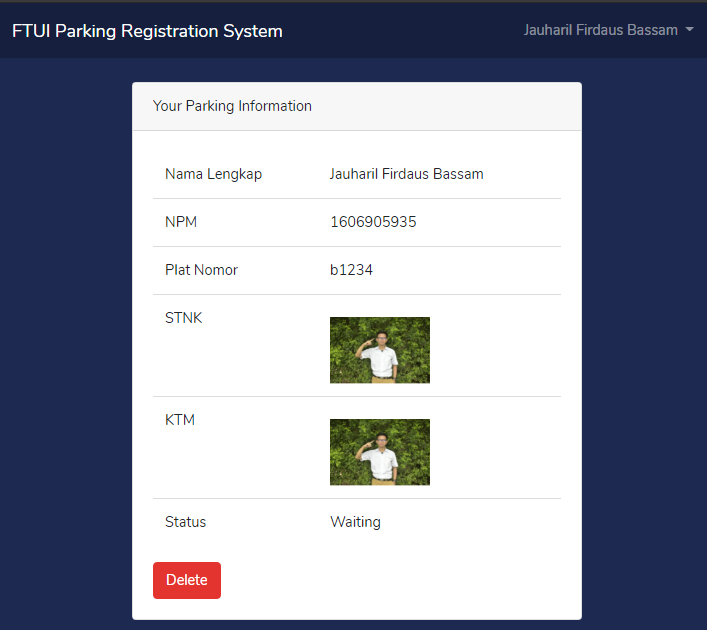


If your sso account not logged in before, you will redirect to the sso login page. You can fill that with your sso username and password

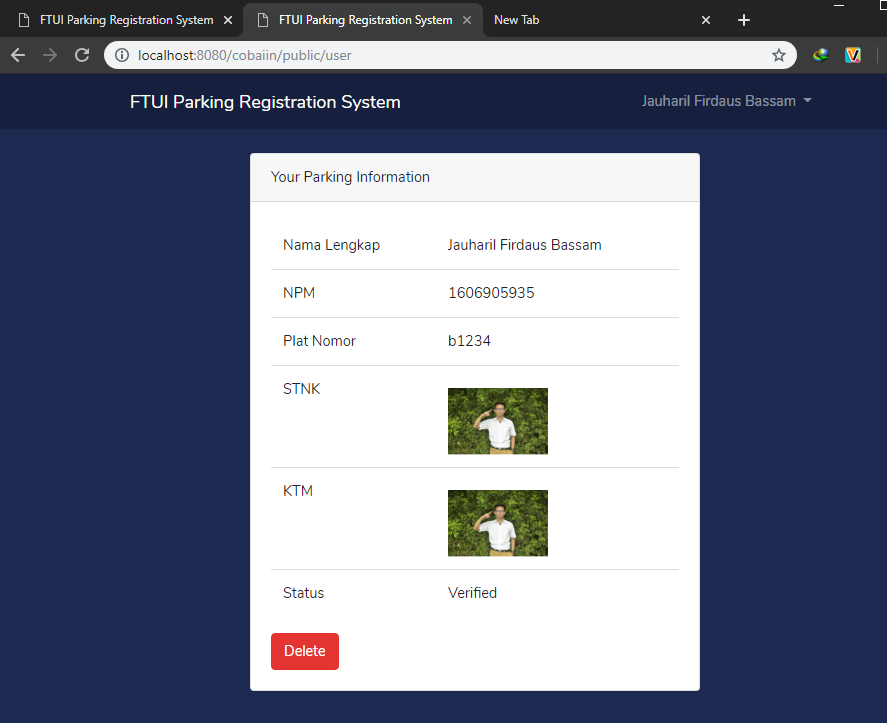
### **6.1.5.4 User mode: information page after login**



You will get this view after login sso. You can see that the information page still empty because you not yet fill form to register.



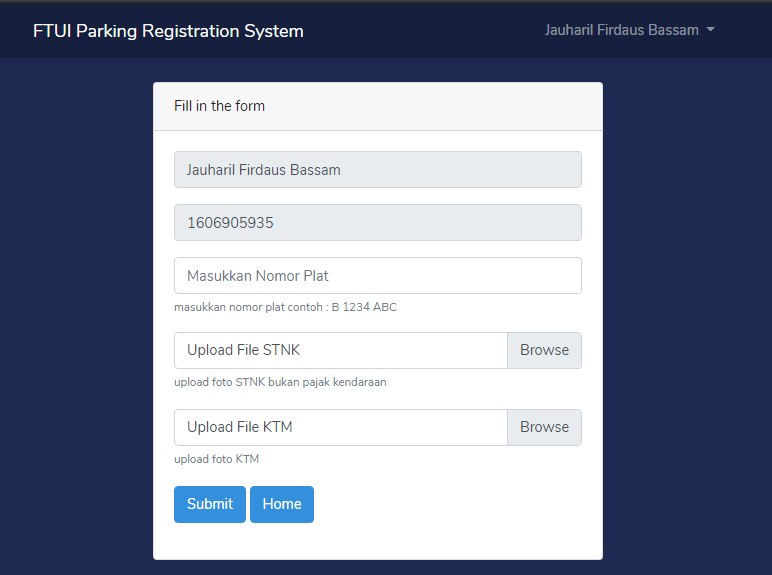
This is the webpage view after you fill the form, status section still “Waiting” because not yet verified by the admin



This is the view when request is verified by admin, you can see “verified” at status section.

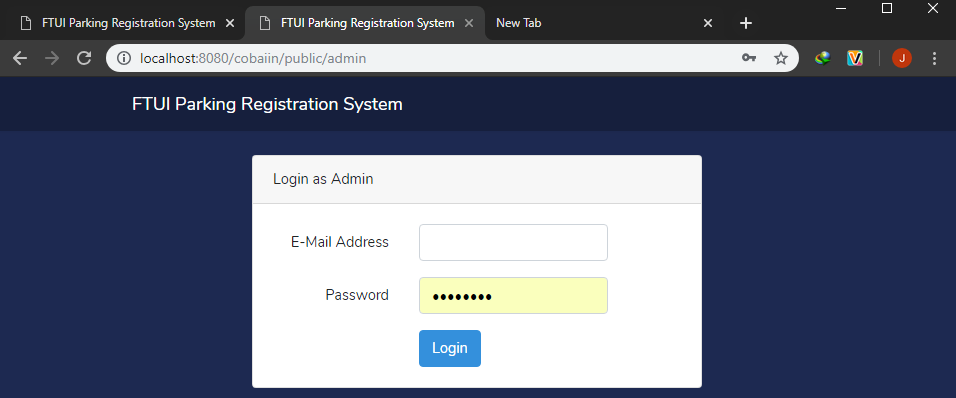
### **6.1.5.5 fill form page**

In this page you can fill yout vehicle number and upload stnk and ktm.



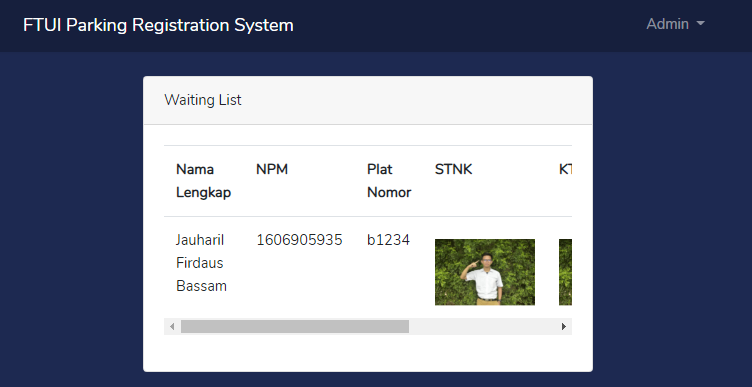
### **6.1.5.6 Admin mode: Login page**

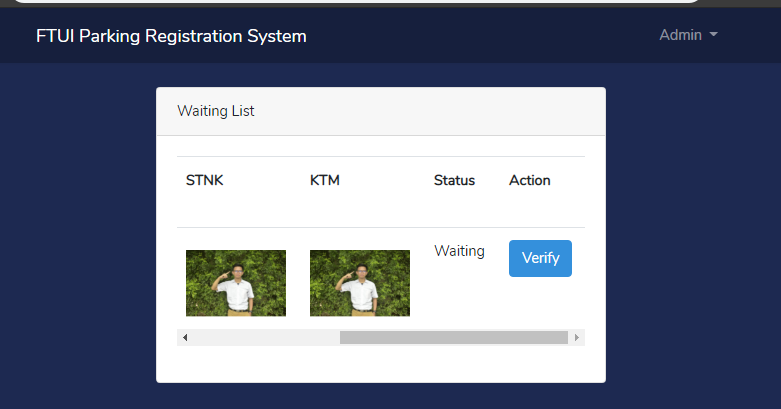
This page is used by admin to login and verify request. Email address and password is give by us to admin so nobody knows.

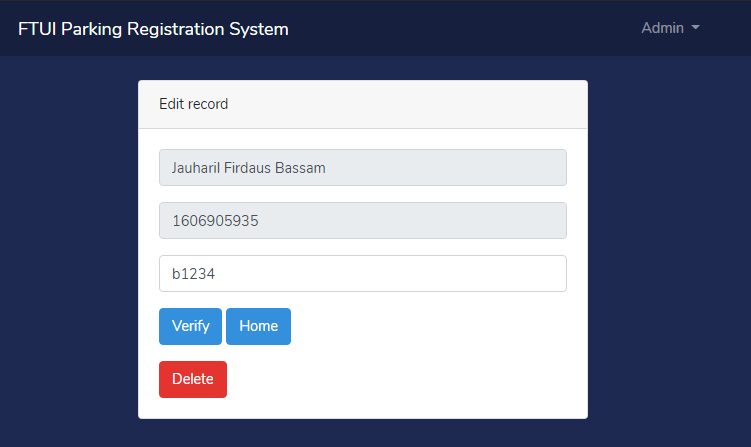


### **6.1.5.7 Admin mode: Home page**

In this page admin can see the request and contains information about name, npm, vehicle number, and stnk and ktm file. After that, admin can click verify button to verifying that request.

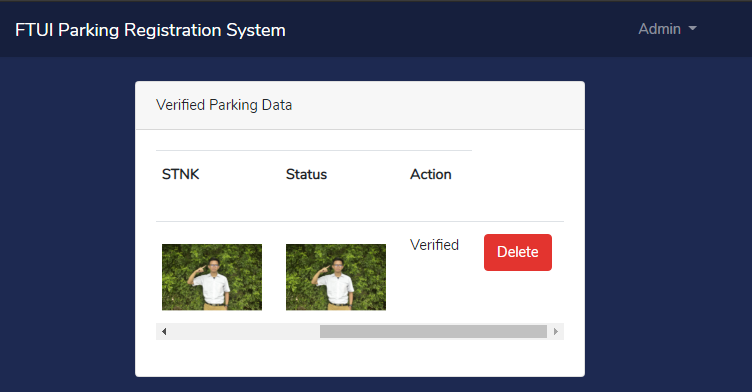






### **6.1.5.8 Admin mode: verified data page**

This page contain information about all request that had been verified by admin.



### **6.1.5.9 Admin mode: Register another admin account page**

This page is used for make another admin account, so the verifying task can be finished faster because helped by another admin.

