Section 1 - Project Description

1.1 Project

PlateMate

1.2 Description

PlateMate serves as a platform that helps colleges and universities manage and improve parking permits for their users. We aim to digitize parking permits by collecting user information in a centralized database and providing enforcement through our license plate scanner web app. The scanner will scan a license plate and notify the parking enforcer whether the car is registered for a valid parking permit or not.

1.3 Revision History

Date	Comment	Author
Tbd	None - Project still in first phase of development	

Contents

Section 2 - Overview

2.1 Purpose

The purpose of this project is to provide a service that takes care of parking permits for an organization. PlateMate will allow users to register for a digitized parking certificate that will be then verified by their organization's parking enforcement through our web app. Our intended audience for our service is colleges and universities, as we plan to increase their user retention. Additionally, our secondary audience is those who purchase and verify parking permits within these institutions, and improve their quality of life when it comes to the stresses of handling physical parking permits.

2.2 Scope

The proposed scope of PlateMate can be divided into two sub-modules. Where there exist two different levels of access: user and parking enforcement.The user is responsible for registering to an academic institution of their choice, given they are attending said institution. After successful registration, the user is then prompted to pay which will result in successfully adding that permit to their account. The parking enforcer is a profile that allows them to scan license plates to verify whether they are registered for their institution's parking permit, as this information is gathered when a new user registers.

2.3 Requirements

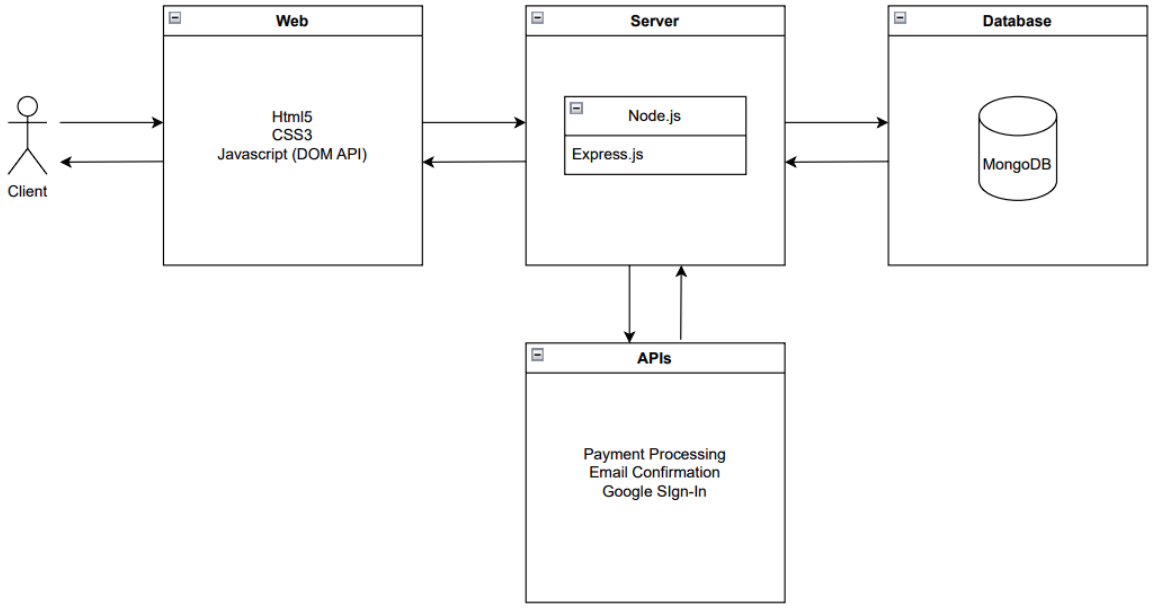
2.3.1 Estimates

#	Description	Hrs. Est.
Num	**Brief description of task / module with link**	**# est**
1	Create Landing Page - The page that visitors first land to.	2
2	Create Server/Database - Establish a server and database connection	5
3	Create Login Page - The page where users log in	1
4	Test Login Page - Test if a valid account is able to log in	2
5	Create email verification - Sends email to tell user they have created an account	4
6	Create Register Page - The page where user's register	1
7	Test Register Page - Test if a valid user can register	1
8	Create Payment Page - The page where the user pays	1
9	Test Payment Page - Test if a valid user can pay	1
10	Create Parking Enforcer Page - The page that is used for scanning license plates	2
11	Test Parking Enforcer Page - Test if the license plate scanner works accordingly	2
12	Connect APIs - Ensure client data gets a response from APIs	7
13	Test APIs - Ensure all APIs are working as intended	3
14	Create Figma UML - A mockup of what the responsive web app will look like.	5
	TOTAL:	37

2.3.2 Traceability Matrix

SRS Requirement	SDD Module
Req 1 - User Interface	7.1, 7.2, 7.3
Req 2 - Hardware	2.2, 7.3, 8, 9
Req 3 - Software	5.1, 5.2
Req 4 - Communication	6.3, 6.4
Req 5 - Non-functional	6.3, 8.1, 9

Section 3 - System Architecture



This module helps paint a picture of what software components are needed and how they interact with each other. The client will interact with the website where it will be built using Html5, CSS3, and Javascript along with the DOM API. When a query is made, the website will first transfer data through the server. The server will be built using node.js along with the express.js module. If the query has to do with a payment or email confirmation, then the server will interact with the listed APIs. The server will also interact with the database where MongoDB will be used. Whether it be fetching, adding, or modifying records in the database.

Section 4 - Data Dictionary

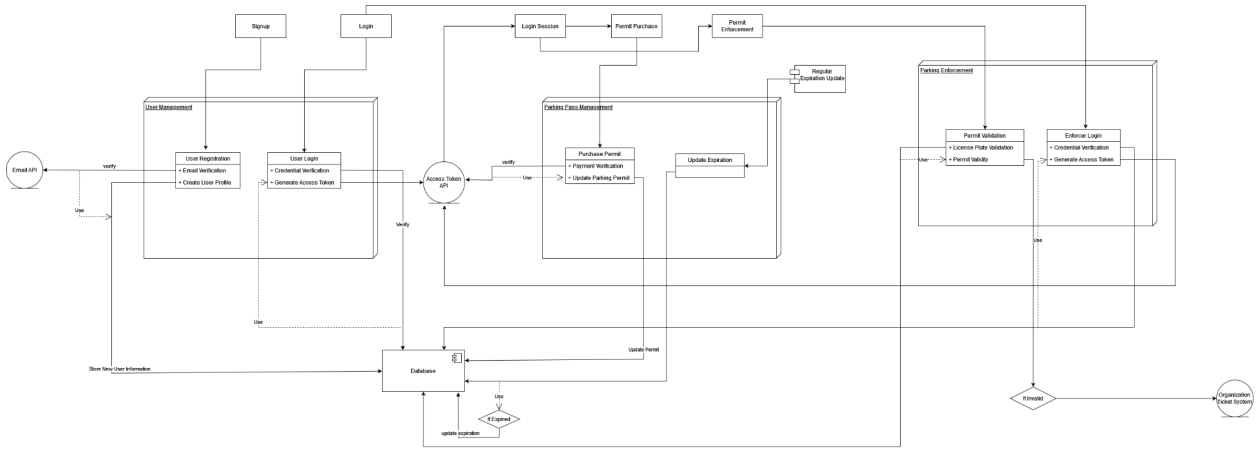
User		
Field	**Notes**	**Type**
_id	Unique identifier	ObjectId
first_name	The user's first name	String
last_name	The user's last name	String
email	The user's email address	String
password	The user's password	String
plate_number	The user's license plate	String
permit	Object containing permit expiration and organization	Object

Enforcer		
Field	**Notes**	**Type**
_id	Unique identifier	ObjectId
username	The parking enforcer's username	String
password	The admin's password	String
organization	Id referencing the enforcer's affiliated organization	ObjectId

Organization		
Field	**Notes**	**Type**
_id	Unique identifier	ObjectId
name	The name of the organization	String

Section 5 - Software Domain Design

5.1 Software Application Domain Chart



5.2 Software Application Domain

5.2.1 User Management

This domain focuses on user registration and authentication.

5.2.1.1 Database

Stores user-related information and makes a connection to their associated organization.

5.2.1.2 User Registration

Use data obtained from frontend to authorize users and store relevant information

5.2.1.2.1 Email Verification

Check if the provided email is a valid address and send a confirmation email.

5.2.1.2.2 Create User Profile

Once the user's email has been verified, add a User entry to the database and store all other relevant information.

5.2.1.3 User Login

Use the provided data from the frontend to verify login credentials and start a session.

5.2.1.3.1 Verify Credentials

Check if the provided credentials match those stored in the database.

5.2.1.3.1 Generate Access Token

Once a user's credentials have been verified, a token can be generated for secure API interactions

5.2.2 Parking Pass Management

This domain is responsible for managing parking pass purchases and expirations

5.2.2.1 Database

Check pass expiration and update if either it has been purchased/renewed or expired

5.2.2.2 Purchase Parking Permit

Verify the provided payment information, if it valid the user's parking permit should be updated

5.2.2.2.1 Verify Payment

Verify and process payment using an external API

5.2.2.2.2 Update Parking Permit

After payment has been verified and processed, the update the user's permit record

5.2.2.2 Permit Expiration

Regularly check the expiration date of users' parking permits and update them if they have already expired.

5.2.3 Parking Pass Enforcement

This domain is responsible for the login of parking enforcers as well as verifying if cars have a valid parking permit. Due to the scope of this project, there will be no enforcer registration, these entries will be manually input into the database.

5.2.3.2 Database

Store login credentials for parking enforcers and access parking permits through license plates.

5.2.2.3 Enforcer Login

Use provided login credentials to authorize parking enforcers and initialize login sessions.

5.2.2.3.1 Verify Credential

Check if the provided credentials math those that are stored in the database

5.2.2.3.2 Generate Access Token

Once an enforcer's credentials have been verified, a token can be generated for secure API interactions

5.2.2.3 Permit Validation

Look up a car's license plate to check whether it has a valid parking pass. Due to the current scope, parking tickets will be managed by the organizations.

5.2.2.3.1 License Plate Validation

Look up a car's license plate and check whether or not it exists in the database.

5.2.2.3.2 Permit Validity

If a license plate is found on the database, the associated permit information will be returned. If either the pass is expired or if it belongs to the wrong organization, then the parking enforcer will be notified that the car does not have a valid parking permit. Afterwards they can use whatever system that they have for handing out tickets.

Section 6 – Data Design

6.1 Persistent/Static Data

6.1.1 Dataset

The logical data model and/or entity relationship diagrams for the persistent and static data are represented in this section. The relationship between key entities and datasets will be illustrated here.

- User: This entity represents information about the users.
  - Attributes: First Name, Last Name, Email, Password, Permit
  - Relationships: Associated with the Organization entity. The permit is tied to an organization
- Enforcer: Represents information about parking enforcers
  - Attributes: Username, Password, Organization
  - Relationships: Associated with the Organization entity. The enforcer works for an organization
- Organization: Contains information about organizations.
  - Attributes: Name
  - Relationships: Connected to both User and Enforcer entities.

6.1.2 Static Data

Static data refers to reference data that does not change frequently. Organizations are currently the only data that falls under this category.

- Organization Static Data:
  - Organizations: Fresno State, Fresno City College, Fresno Pacific University

6.1.3 Persisted data

Persisted data is information that is stored in a structured manner inside of a database. This would include everything currently under the Dataset subsection.

6.2 Transient/Dynamic Data

Transient or Dynamic data is data that changes frequently and often is not stored in a persistent manner. Data that would fall under this section would include access tokens used for authentication.

- User:
  - Verification token received when creating an account
- Organization:
  - Tokens received that allows for the creation of Enforcer accounts

6.3 External Interface Data

External interface data refers to data that is exchanged between external systems. This would include information obtained from external APIs for payment processing, Google login, and email verification.

- Visa:
    - Payment API allowing for safe encrypted payment format.
- Google:
    - Account creation through Google account link with Google API.

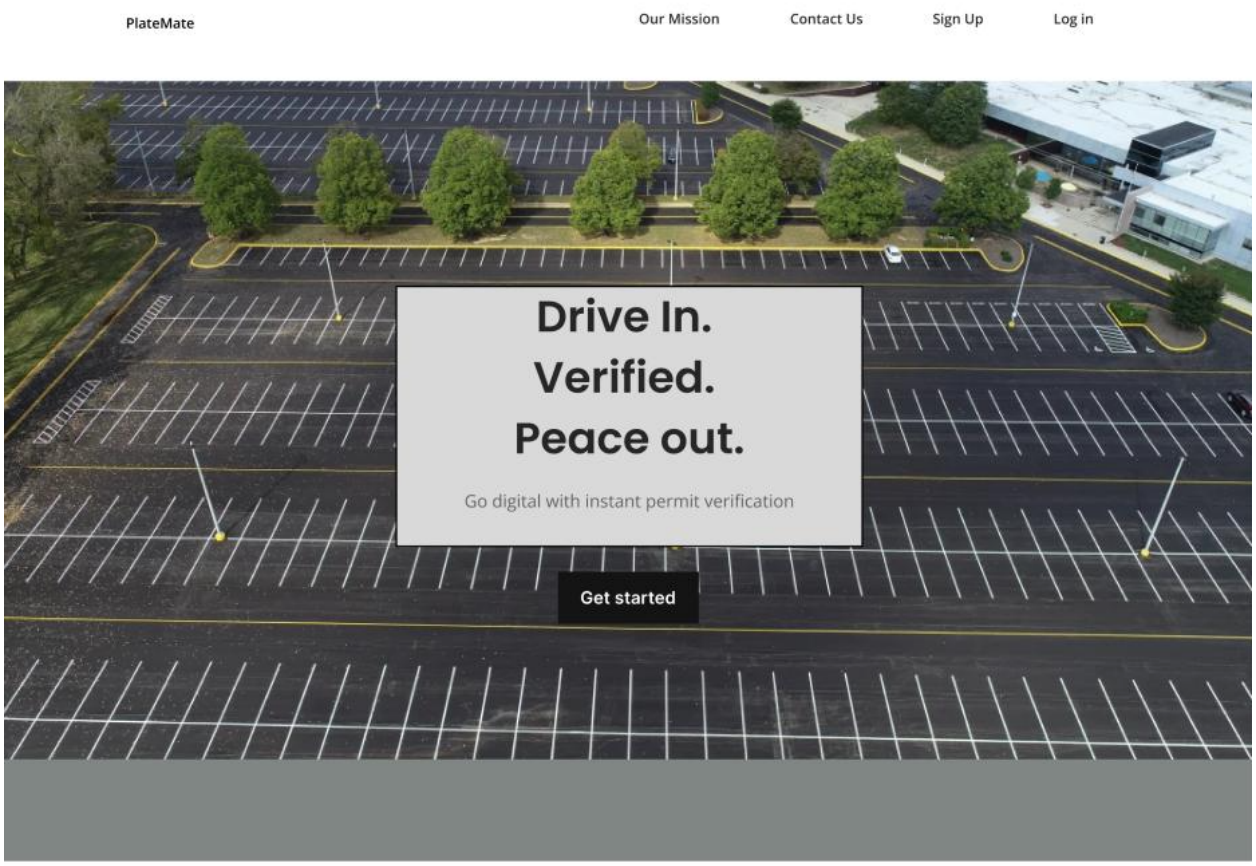
6.4 Transformation of Data

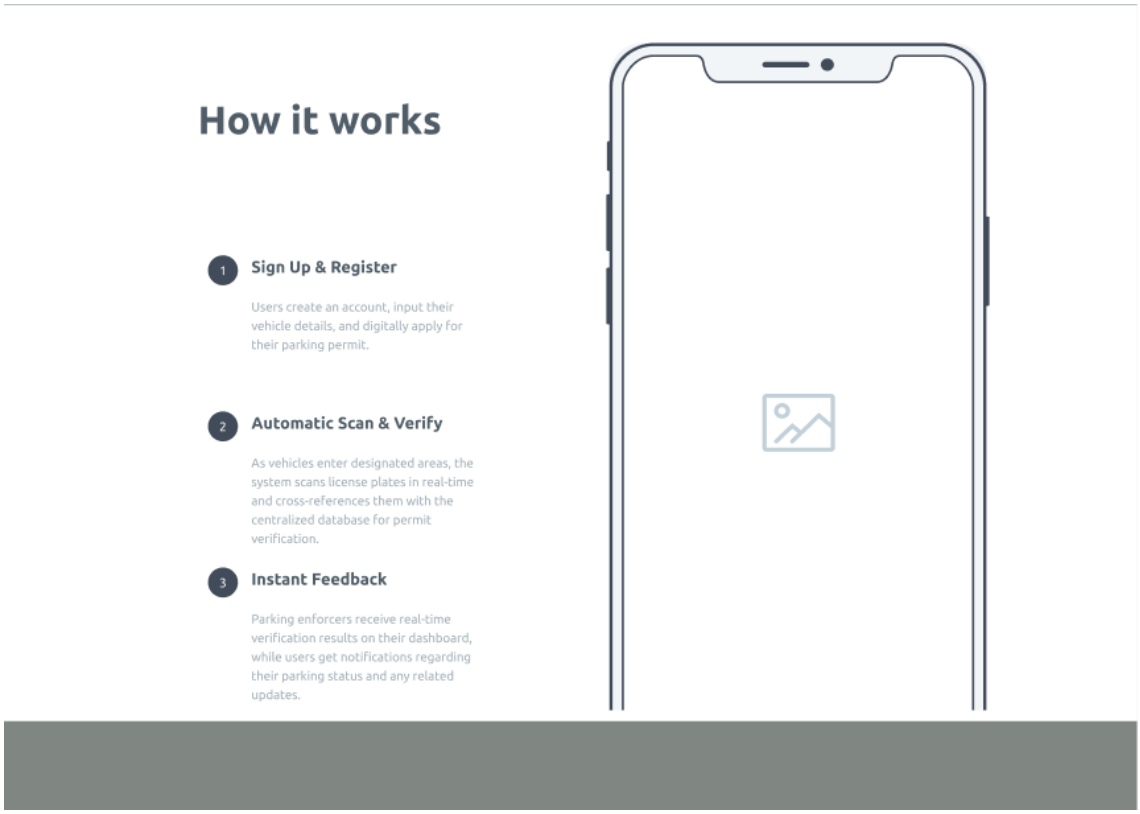
Data that is transformed from one format to another. Within the current scope there is no data being transformed

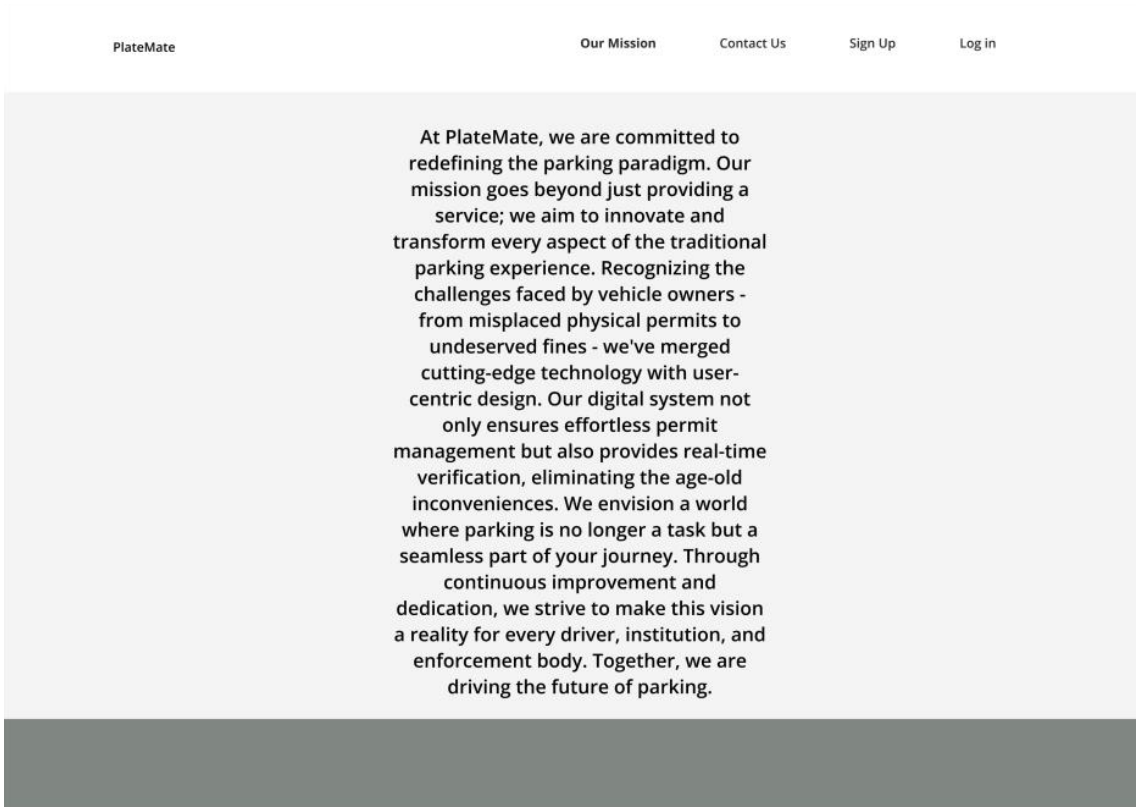
Section 7 - User Interface Design

7.1 User Interface Design Overview

The screenshots below display the mockups created through Figma.



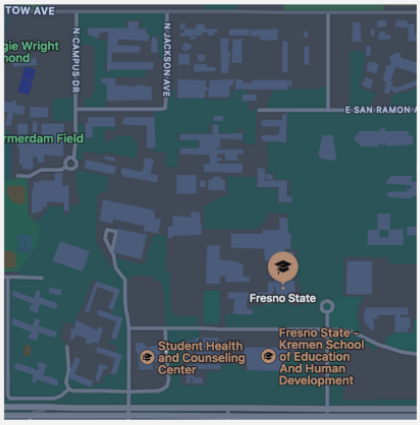

How it works
1
Sign Up & Register
Users create an account, input their vehicle details, and digitally apply for their parking permit.
2
Automatic Scan & Verify
As vehicles enter designated areas, the system scans license plates in real-time and cross-references them with the centralized database for permit verification.
3
Instant Feedback
Parking enforcers receive real-time verification results on their dashboard, while users get notifications regarding their parking status and any related updates.


PlateMate
Our Mission
Contact Us
Sign Up
Log in
At PlateMate, we are committed to redefining the parking paradigm. Our mission goes beyond just providing a service; we aim to innovate and transform every aspect of the traditional parking experience. Recognizing the challenges faced by vehicle owners - from misplaced physical permits to undeserved fines - we've merged cutting-edge technology with user-centric design. Our digital system not only ensures effortless permit management but also provides real-time verification, eliminating the age-old inconveniences. We envision a world where parking is no longer a task but a seamless part of your journey. Through continuous improvement and dedication, we strive to make this vision a reality for every driver, institution, and enforcement body. Together, we are driving the future of parking.

Contact Us

PlateMate Our Mission Contact Us Sign Up Log in

Contact us



We'd love to hear from you

We're always eager to hear from you! Whether you have queries, feedback, or simply wish to learn more about our revolutionary parking solutions, don't hesitate to reach out. At PlateMate, every voice counts, and we're committed to ensuring a seamless experience for all. Get in touch and let's drive the future of parking together!

5241 N Maple Ave

559-278-4240

PlateMate@gmail.com

Sign up

PlateMate Our Mission Contact Us Sign Up Log in

Sign Up. Park Easy.

First Name:

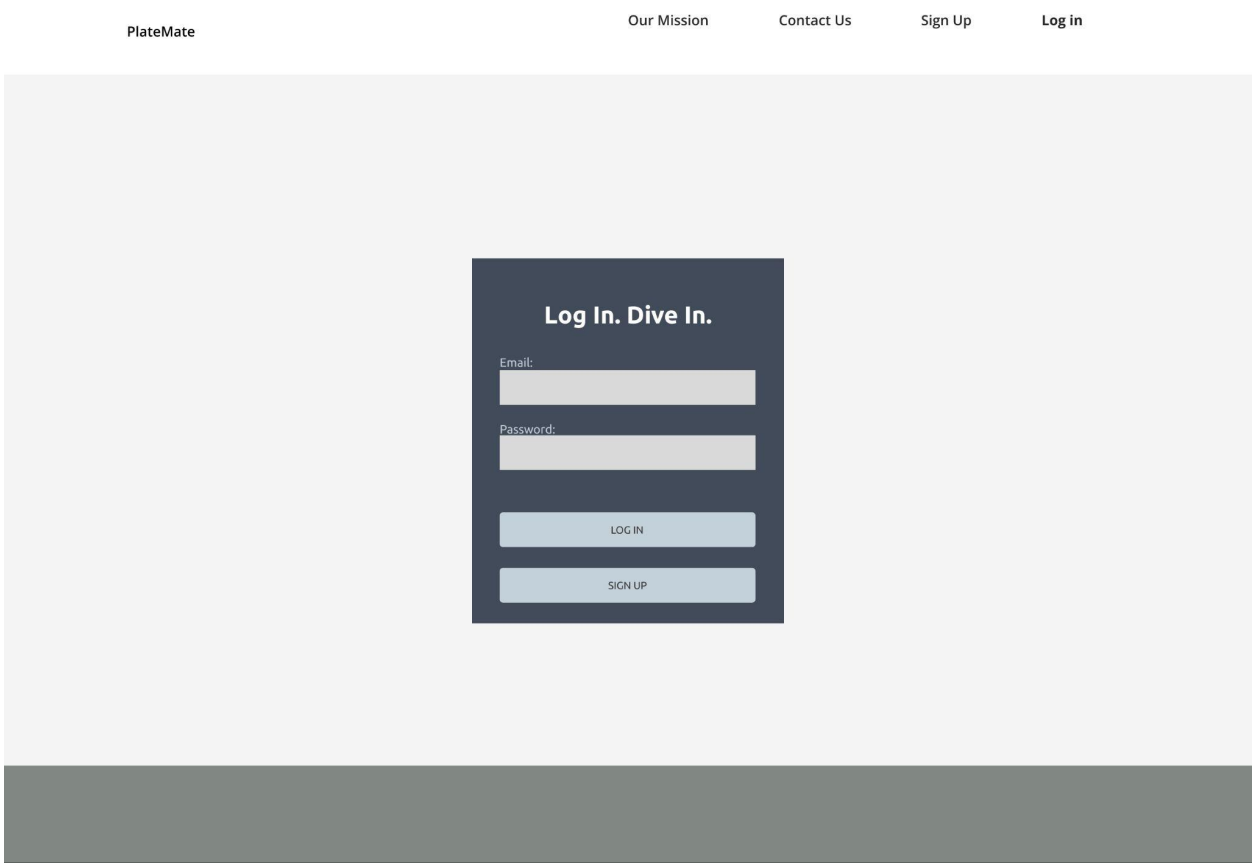
Last Name:

Email:

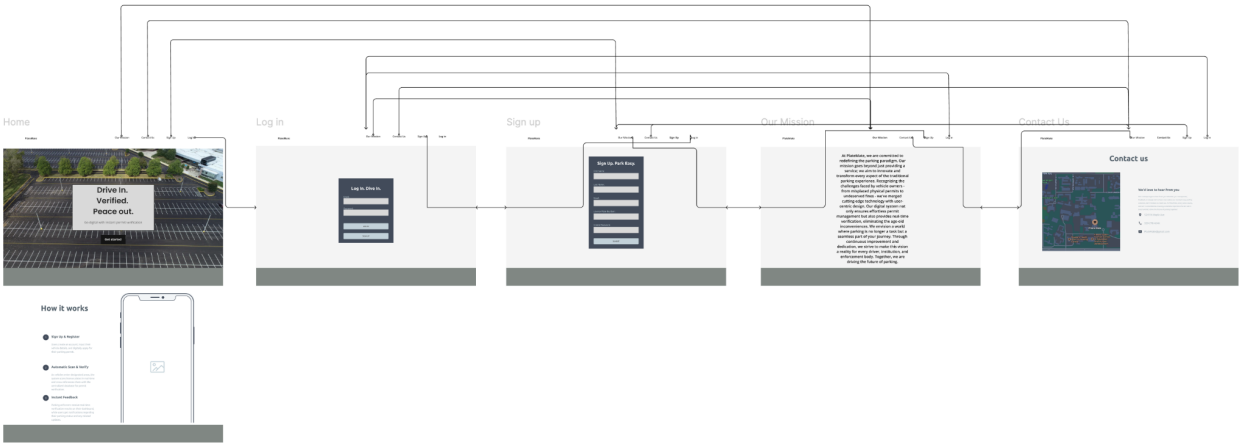
License Plate Number:

Create Password:

SUMBIT


PlateMate
Our Mission
Contact Us
Sign Up
Log in
Log In. Dive In.
Email:
Password:
LOG IN
SIGN UP

7.2 User Interface Navigation Flow



Homepage:
The homepage serves as the gateway to our website. At the top, users can easily access the "Log In", "Sign Up", "Contact Us", and "Our Mission" pages through the navigation bar. Clicking on "Our Mission" directs users to a page detailing our platform's objectives and values, while the other options lead to their respective functionalities. Each page also contains navigational elements, ensuring users can seamlessly move between different sections of our site.

Login Page:
The "Log In" page provides a secure entry point for returning users. With clearly marked fields for username and password, the process is designed for speed and ease. The page also features the slogan "Log In. Dive In.", emphasizing immediate access to the platform's features. From here, users can navigate back to the homepage or access other sections like "Sign Up", "Contact Us", or "Our Mission".

Sign Up Page:
Tailored for newcomers, the "Sign Up" page facilitates the creation of a new user account. It offers a structured form for entering personal details and vehicle information. The user-friendly interface, paired with the slogan "Sign Up. Park Easy.", underscores the platform's promise of simplicity and convenience. Links to other major sections such as "Log In", "Contact Us", and "Our Mission" are also accessible, ensuring users can easily switch between actions.

Our Mission Page:
This section delves into the core philosophy and objectives behind the PlateMate system. Users can immerse themselves in the vision and values that drive the platform, gaining a deeper understanding of its purpose. Navigation options at the top allow users to transition smoothly to the "Log In", "Sign Up", "Contact Us", or return to the homepage.

Contact Us Page:
Serving as a bridge between users and the platform, the "Contact Us" page offers multiple channels for communication. An embedded map suggests the company's physical location, while detailed contact information—including an address, phone number, and email—ensures users have various means to reach out. As with other pages, the top navigation bar provides quick access to the homepage, "Log In", "Sign Up", and "Our Mission" sections.

Entry Point - Homepage:

- Initial Interaction: Users are greeted by the platform's core message, encapsulated by the slogan: "Drive In. Verified. Peace out."
- Navigation Bar: Located at the top, providing immediate options for "Log In", "Sign Up", "Contact Us", and "Our Mission".
- User Journey Pathway: By selecting any of the top navigation bar options, users are directed to the corresponding page, with each page allowing a return journey back to the homepage or onward navigation to other sections.

Secure Access - Login Page:

- Primary Interaction: Fields for username and password, emphasizing quick access.
- Feedback Mechanism: "Log In. Dive In." slogan reassures and prompts users about the ease of entry.
- Navigation Options: Links available to either return to the homepage or proceed to "Sign Up", "Contact Us", and "Our Mission".

Registration Gateway - Sign Up Page:

- User Task Flow: Structured form requesting personal and vehicle details.
- Feedback Mechanism: "Sign Up. Park Easy." slogan provides reassurance of the process's simplicity.
- Navigation Options: Easy toggling between the "Log In", "Contact Us", and "Our Mission" sections.

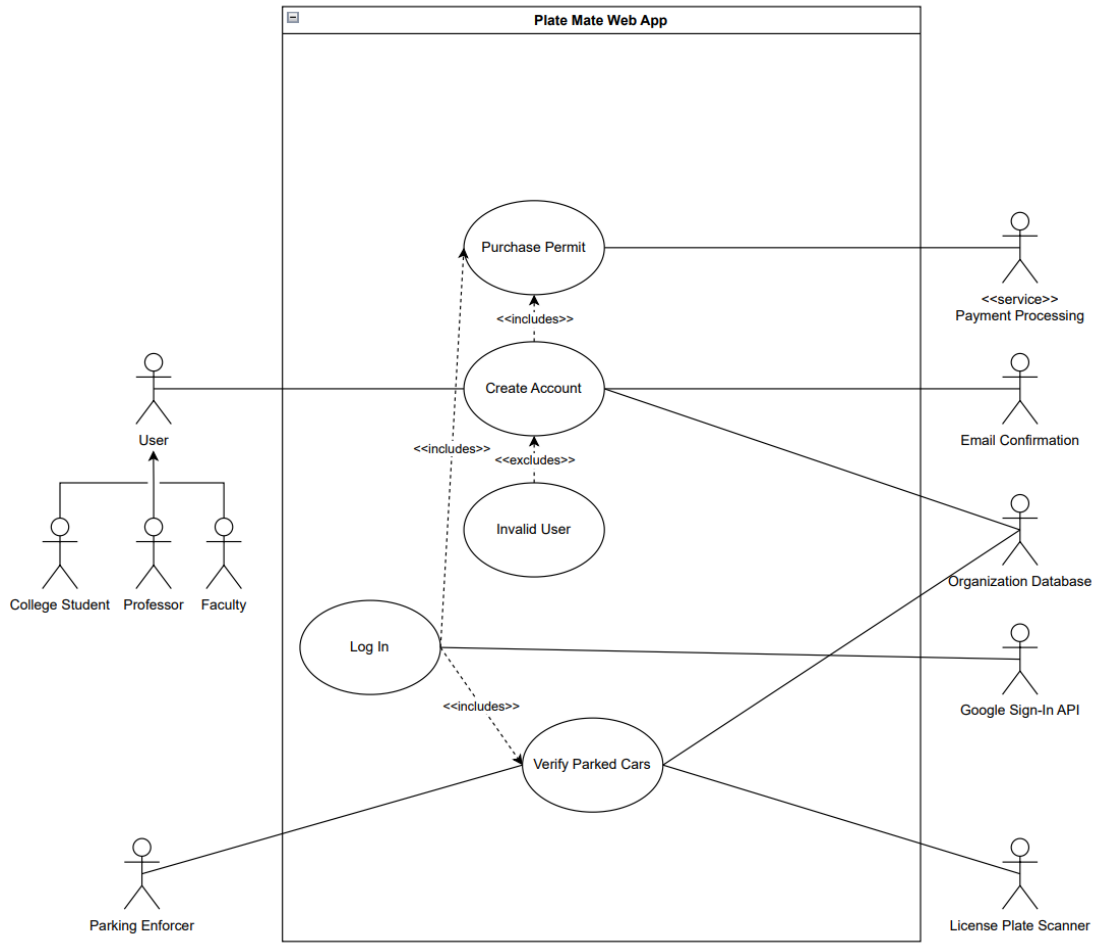
Platform Philosophy - Our Mission Page:

- Informational Content: Detailed overview of the PlateMate system's objectives and foundational values.
- Navigation Options: Seamless transition to the "Log In", "Sign Up", "Contact Us", or a return pathway to the homepage.

Engagement & Outreach - Contact Us Page:

- Interactive Map: Visual representation of the company's physical location.
- Contact Details: Comprehensive list of communication channels, including address, phone number, and email.
- Navigation Options: Quick links at the top, directing users to the homepage, "Log In", "Sign Up", and "Our Mission".

7.3 Use Cases / User Function Description



A user consists of anyone who wants to purchase a parking permit for their college or university of choice. They must create an account where they will have the option to add parking permits to their profile. They can only successfully purchase the parking permit if they belong to the college or university whether they be a student, professor, faculty member, etc. The parking enforcer is the other type of use case, where they must first log into their account to access the ability to verify parked cars.

Section 8 - Other Interfaces

Payment processing API - Allows users to make a payment for their parking permit for the college or university of their choice.

Google account login API - Allows for users to log into our website. This will also help with user account personalization.

Email confirmation API - Allows to verify that a user is affiliated with a certain college or university.

8.1 Interface X

- The interaction of these APIs are with our server.
- These API's follow the REST protocol.
- The message formats will be in the JSON format generally flowing through TCP/IP.
- Failure conditions involve a user entering invalid information.
- All these APIs require handshaking as a connection must be established.

Section 9 - Extra Design Features / Outstanding Issues

After the completion of the service that PlateMate aims to deliver, this table will serve as a list of extra design features to implement in the future.

Extra Features	Overview
Admin Account	An admin account would have the ability to view, modify, or delete user data, according to what organization they have access to.
Picture To Text	This feature would apply to the parking enforcer profile. Instead of typing in license plate numbers, they would instead be able to use their camera to grab text from license plates.

Section 10 – References

Jose's Software Requirement Specification
Rigo's Software Requirement Specification
William's Software Requirement Specification
Youssif's Software Requirement Specification

Section 11 – Glossary

Glossary of terms / acronyms

HTML	Hyper text markup language
QA	Quality Assurance
SDD	Software Design Document
SRS	Software Requirement Specification

Tbd	To be decided
Tbn	To be named
API	Application Programming Interface
REST	Representational State Transfer
JSON	JavaScript Object Notation
TCP/IP	Internet Protocol Suite
Enforcer	An individual companies traffic enforcer

Section 12 – For Grading Purposes

Name	What you did.
Rigo	1.1 - Helped with the project name. 2.1 - Helped write project description. 2.2 - Helped write project scope. 2.3.2 - Helped finish traceability matrix. 7.3 - Helped create the use case diagram. 9 - Helped come up with different extra features and outstanding issues. 10 - Added my SRS. 11 - Added terms.
Jose	1.1 - Helped with the project name. 1.2 - Created project description. 2.1 - Helped write project description. 2.2 - Helped write project scope. 2.3.1 - Added my list of jobs. 3 - Created the system architecture figure and description. 7.3 - Helped create the use case diagram. 8 - Helped describe APIs 8.1 - Helped go in depth on the APIs. 9 - Created and filled table for extra features. 10 - Added my SRS.

	11 - Added terms
William	1.1 - Helped with the project name. 2.3.1 - Populated table with jobs. 2.3.2 - Helped finish traceability matrix. 7.3 - Helped create the use case diagram. 8 - Helped decide which API's 8.1 - Help decide where to implement 9 - Helped come up with different extra features and outstanding issues. 10 - Added my SRS. 11 - Added terms
Youssif	1.1 - Helped with the project name. 7.1 - Created Figma Website Mockup 7.2 - Created and described UI Navigation UML 7.3 - Helped create the use case diagram. 9 - Helped come up with different extra features and outstanding issues. 10 - Added my SRS.
Emanuel	1.1 - Helped with the project name. 4 - Filled out data tables 5.1 - Made domain chart 5.2 - Defined application's software domain 6 - Specified design regarding data 7.3 - Helped create the use case diagram. 9 - Helped come up with different extra features and outstanding issues. 10 - Added my SRS.