DOWLIN INC.

BIKE MEE! MILESTONE

Group A:

Blessing Ajibero

Baihan Li

Emmanuel Akoja

Bader F Syeda

Introduction 2

Stories and Scenarios.. 2

Use Cases 4

System Architecture 5

Entity Relational Diagram 5

UML Model.. 6

Requirements Specification 7

Design Document 8

Project Backlog 14

Sprint Backlog...16

Introduction

Bike-Mee is a web app by Dowling Inc. Dowling Inc. It provides bikes for your convenient rides within a 30-mile radius of the Louisiana State University campus, Baton Rouge. We have the best locations that are easy to find. You can get a bike at the closest dock to your location, take a safe ride to your destination, and return the bike to the dock close to your destination. Our application promises to be user friendly and convenient for all demographics.

Stories or Scenarios

Story 1 (Registered user)

Derick is an undergraduate student of LSU that lives five minutes bike distance away from his class. Although he has a car, he would prefer to ride a bike to the campus because he likes to keep his body in shape and wants to be economical about LSU parking pass expenses. Derick only has one in presence class on campus and sees no point in paying for a parking pass. He also does not own a bike but will like access to a campus ride for his class.

Derrick sees an ad for Dowlin's Inc bike business on social media and sees that using Dowlin's bike on Tuesday at 10:30 am is suitable and economical for him. Dowlin offers dock stations in various locations that are easily accessible to him, and he decides to give it a try.

He would log in to his account every Tuesday, at 6:30 am, then check for the availability of a bike at the nearest dock location to him. If available, he would book to use one at 7:00 am, and the app will charge him $5, starting from 7:00 am. If there is no single bike in the area close to him, he will check for availability at the next closest location to him.

Scenario 1

Initial Scenario: Derrick downloads the Dowlin app. He then registers with his phone number and is quickly sent a code for verification. After this stage, Derick completes his registration by typing in his name, email address, credit card number. Afterward, he has access to the app.

Normal: Derrick uses the app to search for the closest dock station to his apartment. He sees that it is five minutes away and proceeds to the specific location. He chooses one of the six bikes present and scans one. The manager gets a notification that a user wants to use the bike. He then grants the user access after verifying his credit card.

Final Scenario: Derrick's details have been authenticated, and he has been granted access to use the bike. He proceeds to campus and enjoys his trip. He sees that there is a dock station close to his class on campus, which is very convenient for him and returns the bike. Afterward, his credit card is charged.

Story 2 (**New User)**

Myla and her five friends like to hang out on weekends. She invites her friend to go bike riding on Saturday. Since none of them own bikes, Sarah, a Dowling user, suggests that they rent bikes from Dowlin for the day. Myla does not want to be responsible for each Bike, so she informed her friends to register on the Dowling Inc. web app and book a ride with her at the same time. She lets them know the information required to register name, address, birth date, phone number, valid credit info, email address, password.

Four of her friends are successfully registered, as they all provided valid information. But Keith, who could not give the correct credit info, was denied an account on the web app. Hence, she could not ride with the others because the successfully registered users did not want to be responsible for her Bike.

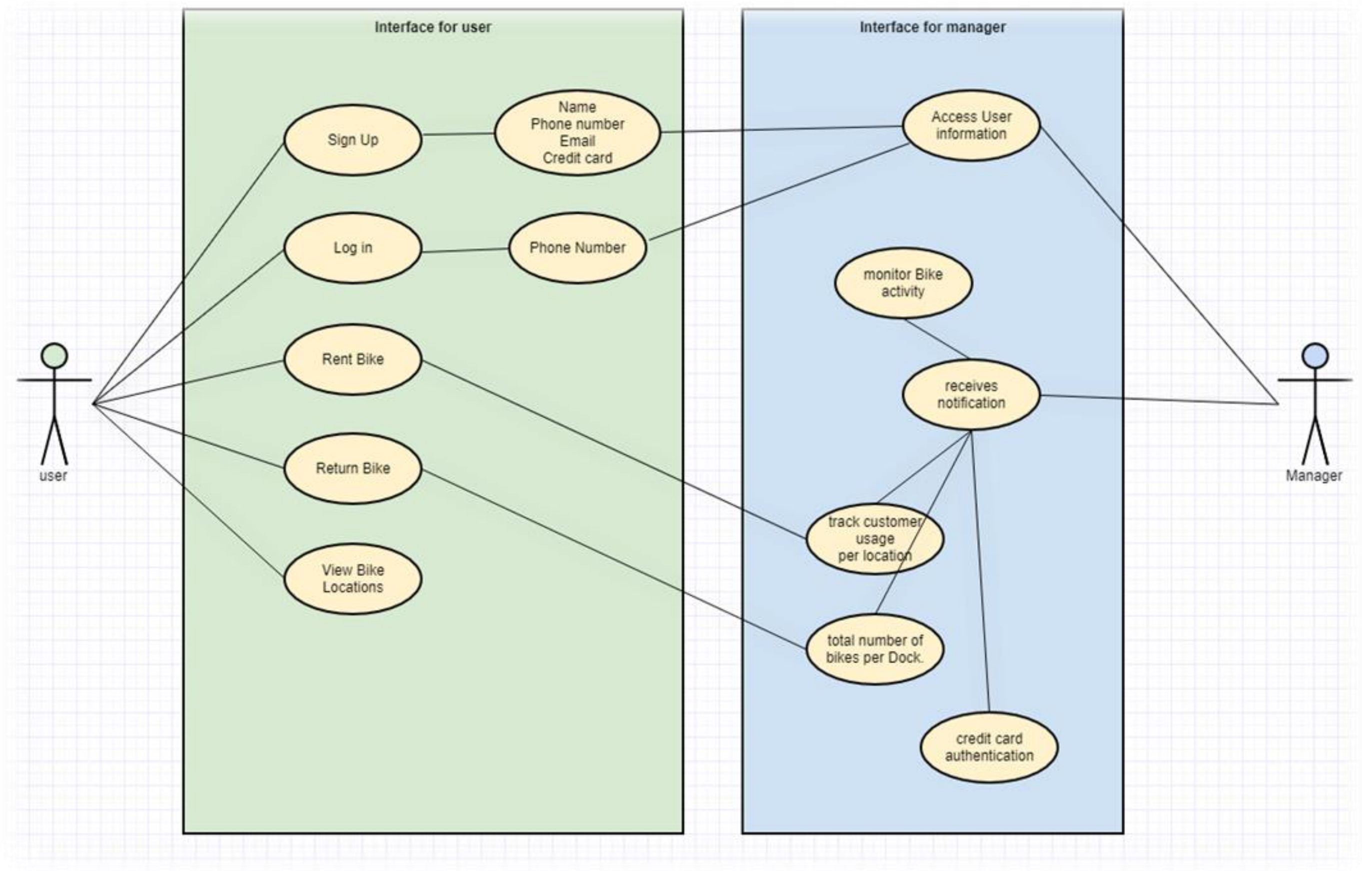
Scenario 2:

Initial Scenario: Myla and her friends decide to meet at a central location to access the same dock station. They agree to meet at the park to go bike riding. Fortunately, Dowlin has a dock station at the park, which is convenient for her and her friends.

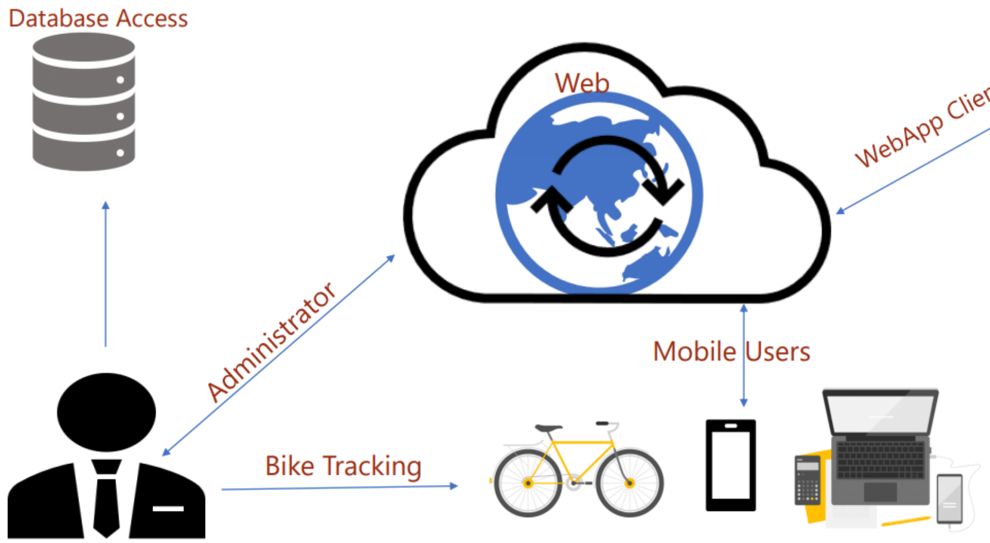
Normal: They meet at the park. Scan their bikes, and everyone except Keith is quickly verified and their credit card information authenticated. The manager gets notified that five bikes have been rented and awaits the notification for their return.

Final Scenario: Myra and her friends enjoyed their ride around the park, however. Sara leaves feedback on the app saying her Bike needs maintenance as it did not move as quickly as the others. The manager notes the location and the Bike. He then schedules it for pick up and maintenance.

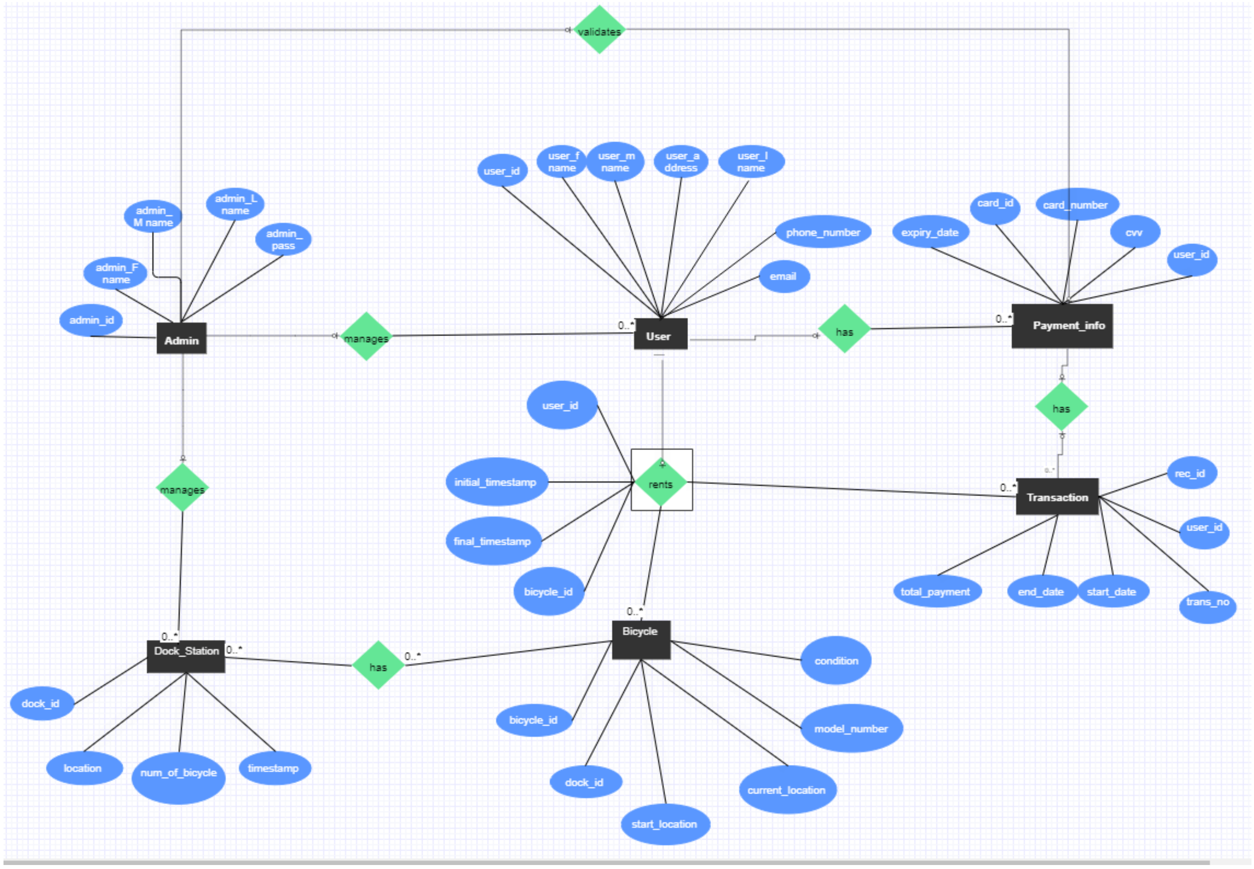
Use cases



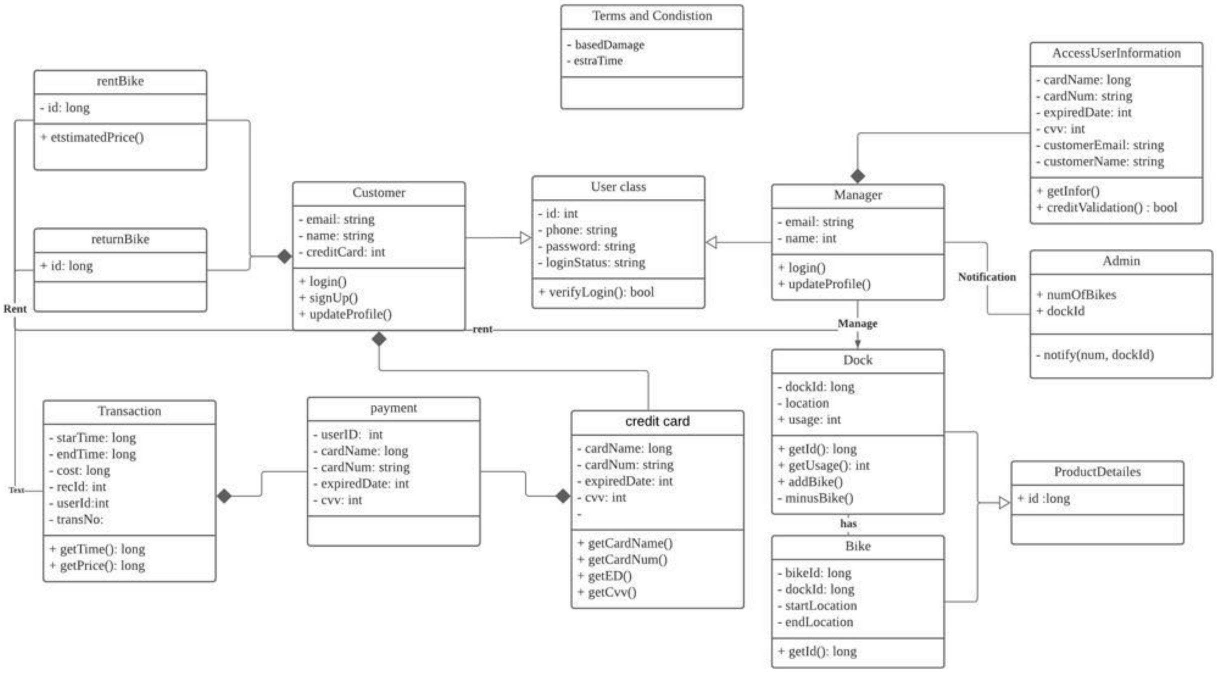
System Architecture



Entity Relationship Diagram



UML Diagram


Terms and Condistion
- basedDamage
- estraTime
AccessUserInformation
- cardName: long
- cardNum: string
- expiredDate: int
- cvv: int
- customerEmail: string
- customerName: string
+ getInfor()
+ creditValidation() : bool
rentBike
- id: long
+ etstimatedPrice()
returnBike
+ id: long
Customer
- email: string
- name: string
- creditCard: int
+ login()
+ signUp()
+ updateProfile()
User class
- id: int
- phone: string
- password: string
- loginStatus: string
+ verifyLogin(): bool
Manager
- email: string
- name: int
+ login()
+ updateProfile()
Notification
Admin
+ numOfBikes
+ dockId
- notify(num, dockId)
Rent
rent
Manage
Dock
- dockId: long
- location
+ usage: int
+ getId(): long
+ getUsage(): int
+ addBike()
- minusBike()
has
Bike
- bikeId: long
- dockId: long
- startLocation
- endLocation
+ getId(): long
ProductDetailes
+ id :long
Transaction
- starTime: long
- endTime: long
- cost: long
- recId: int
- userId:int
- transNo:
+ getTime(): long
+ getPrice(): long
Text
payment
- userID: int
- cardName: long
- cardNum: string
- expiredDate: int
- cvv: int
credit card
- cardName: long
- cardNum: string
- expiredDate: int
- cvv: int
-
+ getCardName()
+ getCardNum()
+ getED()
+ getCvv()

User Requirements Specification

The application would be able to provide services to 60 users in real-time operation.

The application is design to provide a fast response to users, prompt notifications on transactions with a high-speed server to propagate them.

The nonfunctional system requirements

Users include students, faculty, and any eligible member of the community. To be eligible, a new user would have to enter valid details to create.

Both the registration, booking of a bike, check out would take less than 10 seconds each.

The GPS map would load on the web application within 3 seconds with all the docking locations around.

System Requirements Specification

Dock station functional requirements:

- Scanning the bar code after credit card authentication will unlock a bike for the user
- The system will log the user details for the opened bike.
- When returning the bike, the system will match the user details and lock the bike. The system prohibits an unauthorized user from unlocking or lock a bike at the dock station.
- When a user tries to lock another object different from the Dowling Inc. bike, the bike stand won't close.
- A reliable data/network connection at each dock station is available.

Application functional requirements:

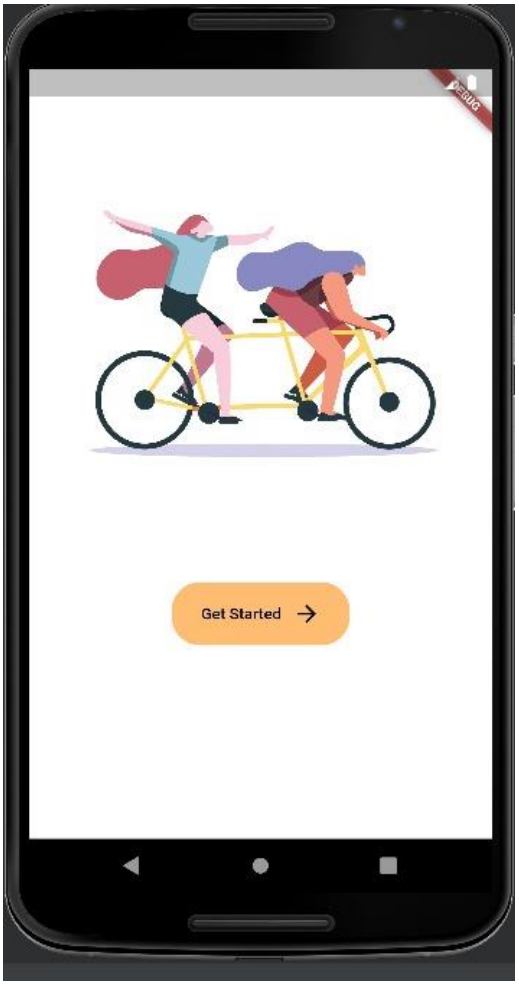
- The application would be able to track the current user of a bike in real-time.
- Users can provide feedback, report faults or any damage on the bike.
- When a bike is reported as faulty by a user, it will disallow such a bike unlock. The user will see "bike needs service."
- The user would be able to view the available number of the bikes at each location.
- The administrator would be able to monitor the system application with high-level details. Remotely enable and disable checkout.
- The administrator will be able to see the bikes that the user reported as faulty.
- After an administrator has repaired a faulty bike, the bike will be available for booking.
- An administrator can disable the use of a bike for maintenance reasons.
- An administrator can monitor the stations' activities, such as view bikes, filter dates, and users.
- An administrator can add, remove a bike from the application.
- An administrator can view, add, edit, and remove users from the backend of the application.

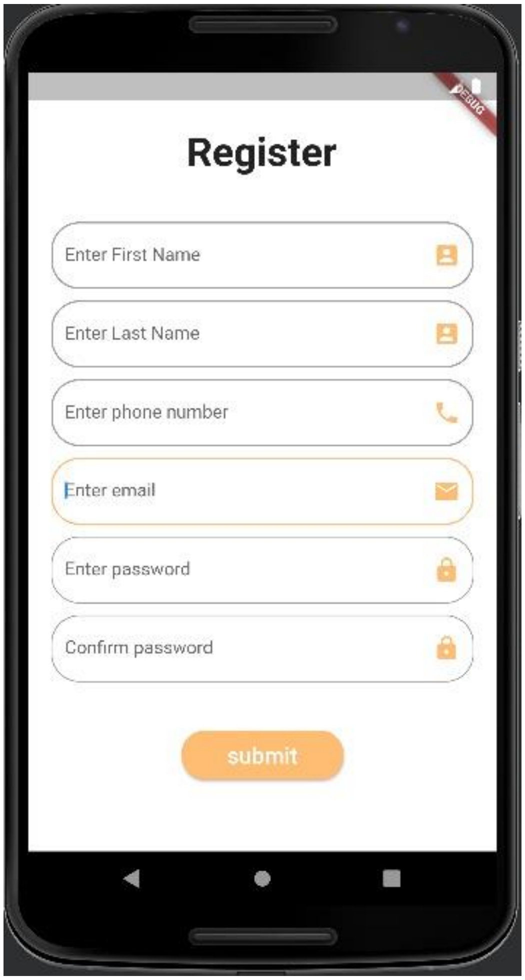
Hardware requirements:

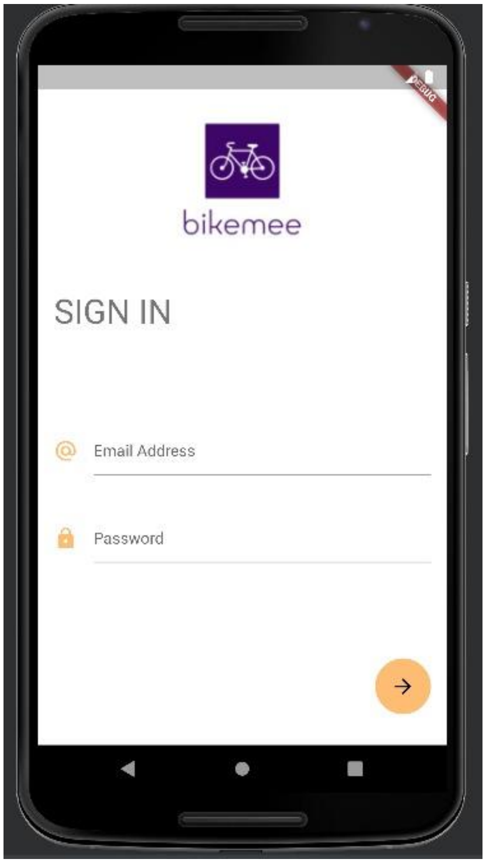
Simplicity and reliability are the priority, with fewer components, maximum functionality at each docking station.

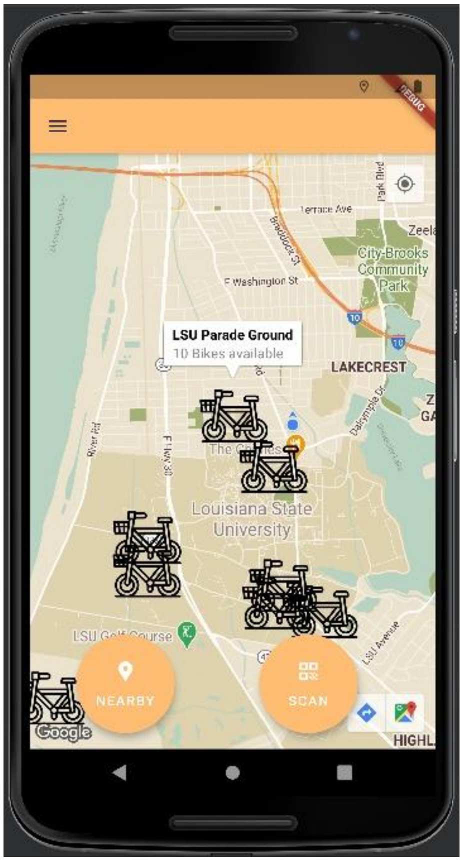
Design Document

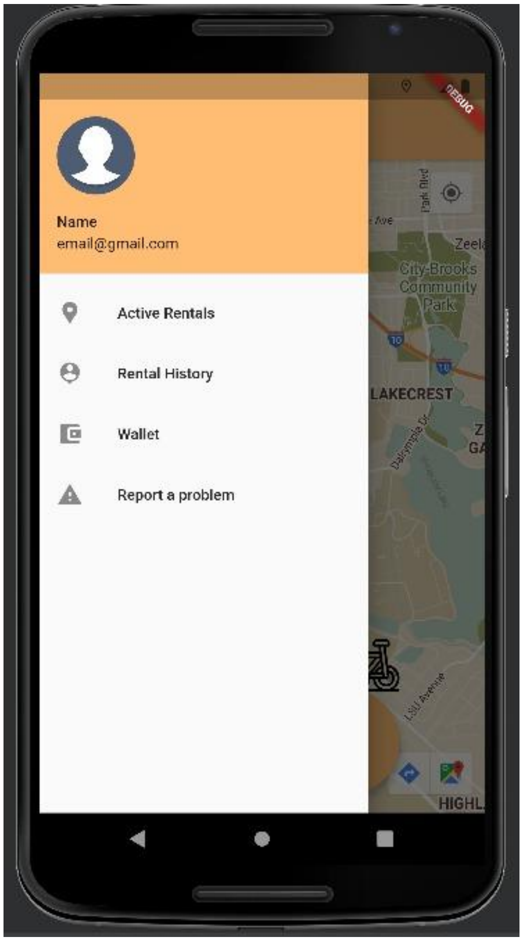
Web App UI

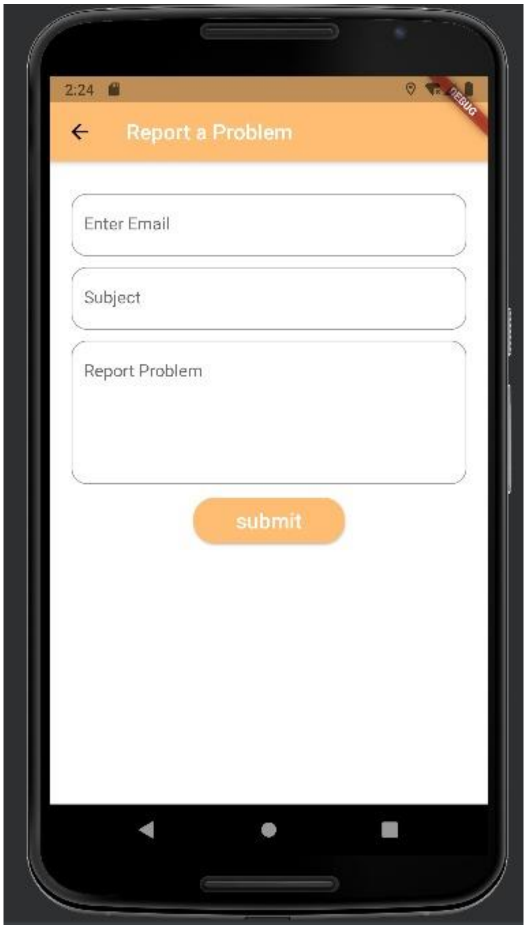


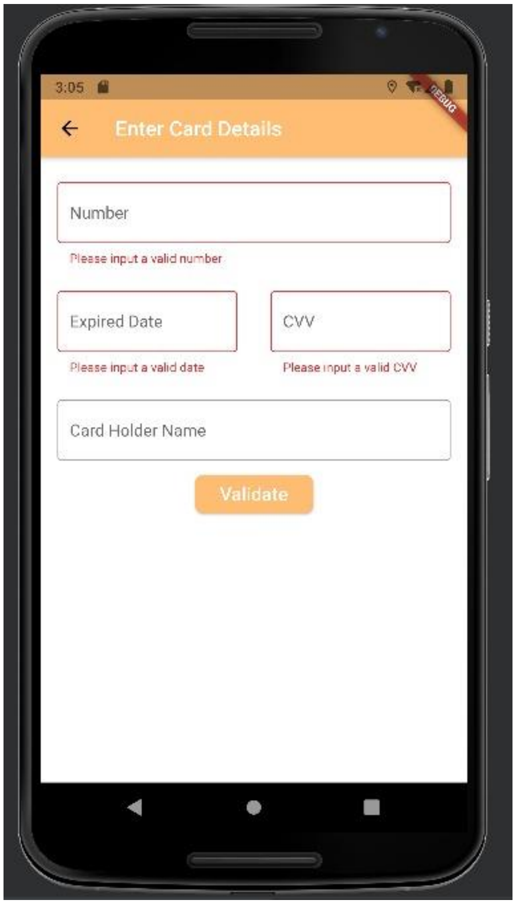



DEBUG
bikemee
SIGN IN
Email Address
Password


DEBUG
Terrace Ave
Park Blvd
Zeela
City-Brooks Community Park
E Washington St
LSU Parade Ground
10 Bikes available
LAKECREST
River Rd
Louisiana State University
LSU Golf Course
LSU Avenue
NEARBY
SCAN
Google
HIGHL


Name
email@gmail.com
Active Rentals
Rental History
Wallet
Report a problem
DEBUG
City-Brooks Community Park
LAKECREST
HIGHL


2:24
DEBUG
Report a Problem
Enter Email
Subject
Report Problem
submit



3:05

Enter Card Details

Number

Please input a valid number

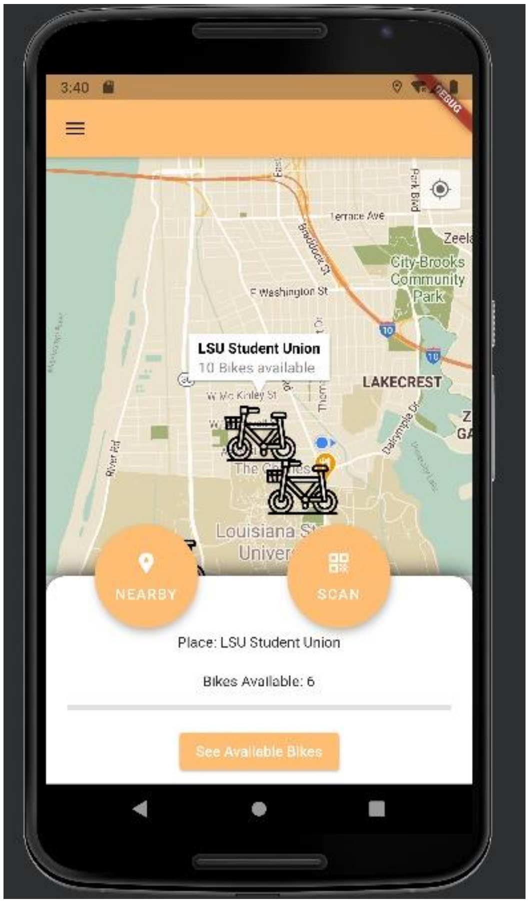
Expired Date

Please input a valid date

CVV

Please input a valid CVV

Card Holder Name

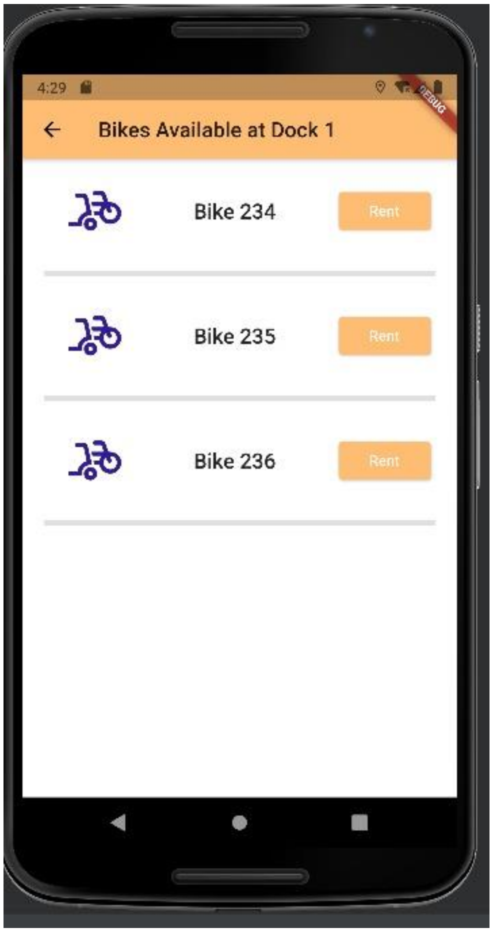


3:40

LSU Student Union
10 Bikes available

Place: LSU Student Union

Bikes Available: 6



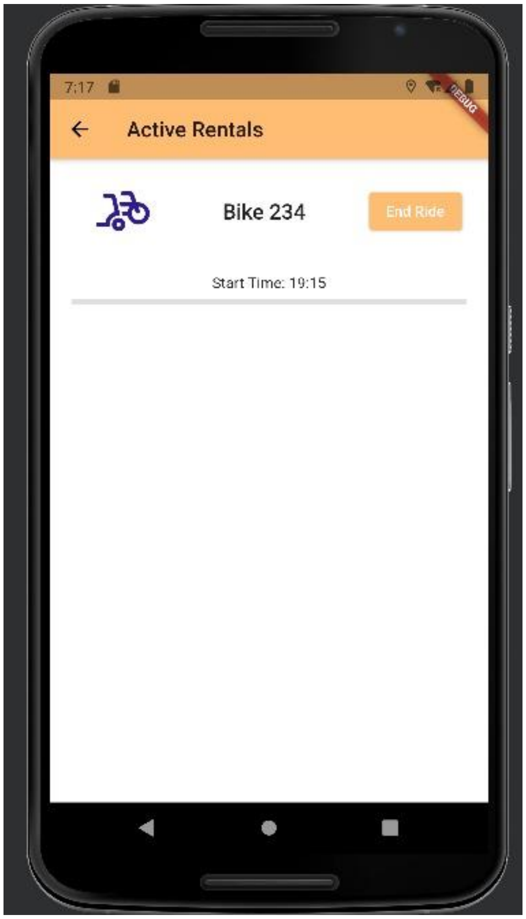
4:29

Bikes Available at Dock 1

Bike 234

Bike 235

Bike 236



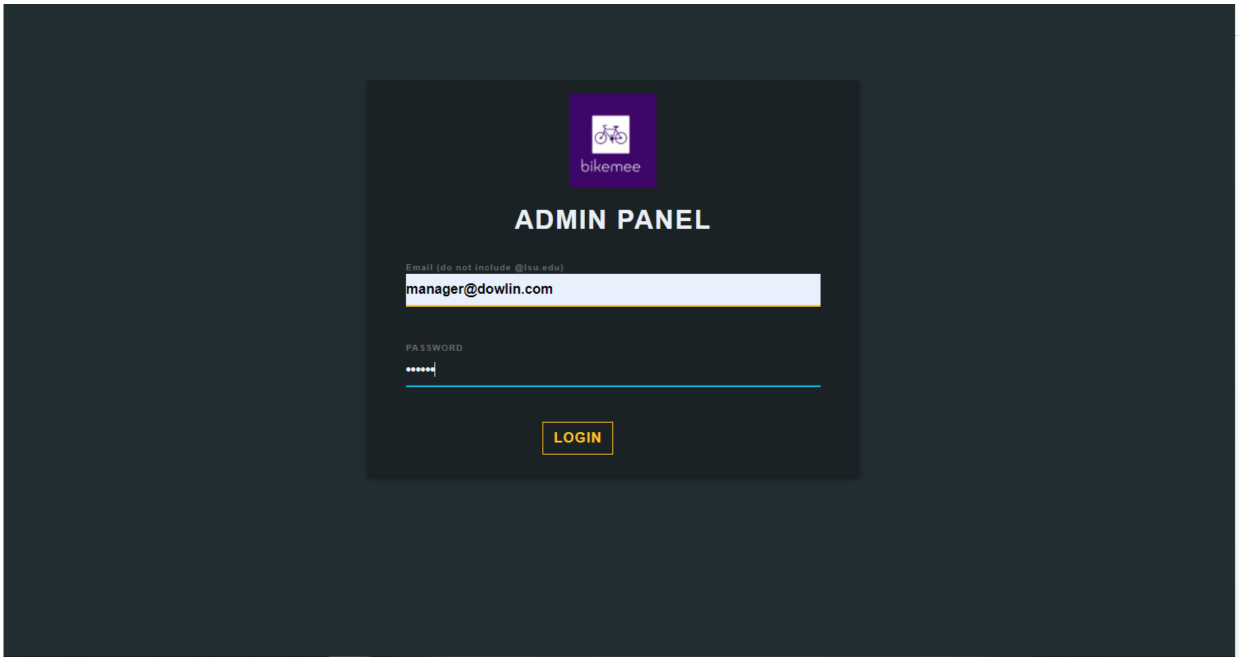
7:17

Active Rentals

Bike 234

Start Time: 19:15

Login for Admin Dashboard



ADMIN PANEL

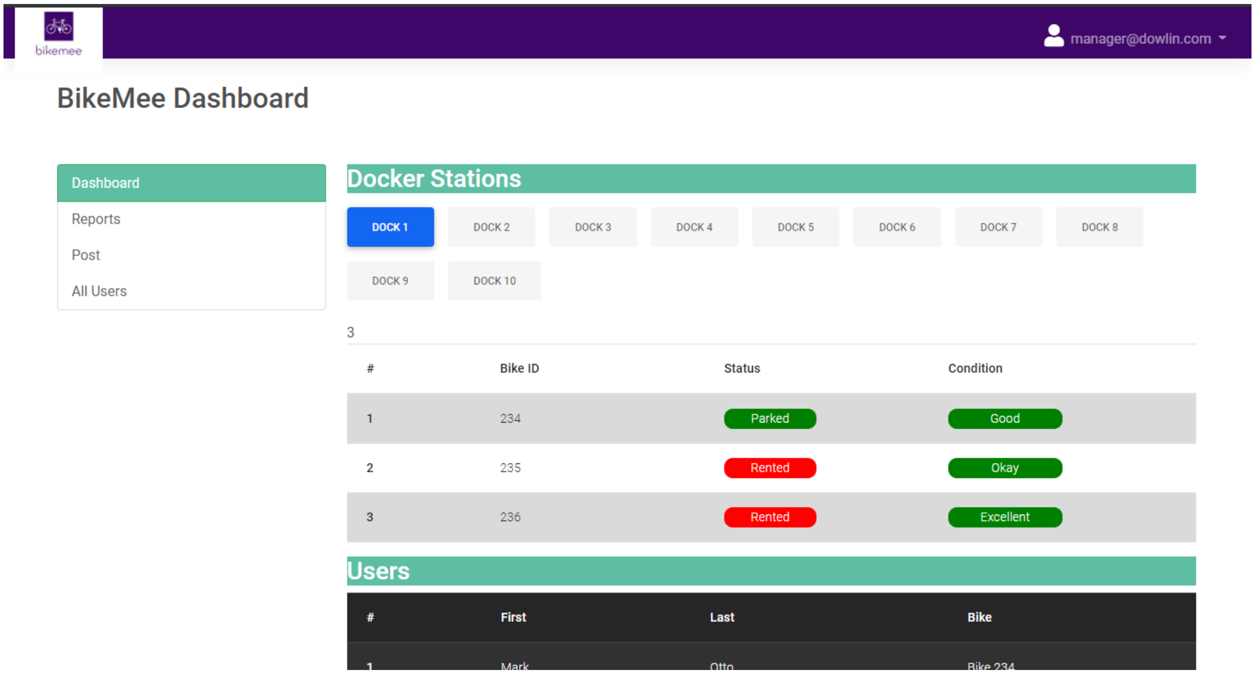
Email (do not include @lsu.edu)

manager@dowlin.com

PASSWORD

••••••

Admin Dashboard



3

#	Bike ID	Status	Condition
1	234	Parked	Good
2	235	Rented	Okay
3	236	Rented	Excellent

#	First	Last	Bike
1	Mark	Otto	Bike 234

Database Update

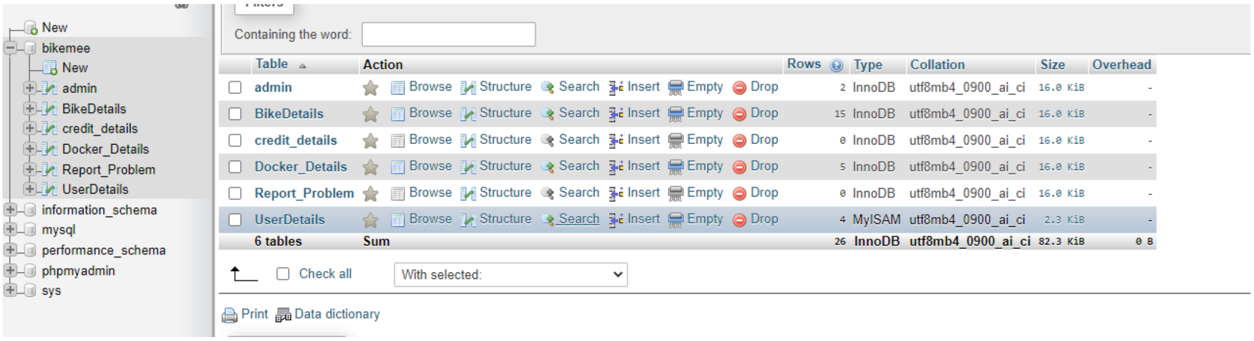


	Table	Action						Rows	Type	Collation	Size	Overhead
□	admin	☆	Browse	Structure	Search	Insert	Empty	Drop	2	InnoDB	utf8mb4_0900_ai_ci	16.0 KiB	-
□	BikeDetails	☆	Browse	Structure	Search	Insert	Empty	Drop	15	InnoDB	utf8mb4_0900_ai_ci	16.0 KiB	-
□	credit_details	☆	Browse	Structure	Search	Insert	Empty	Drop	0	InnoDB	utf8mb4_0900_ai_ci	16.0 KiB	-
□	Docker_Details	☆	Browse	Structure	Search	Insert	Empty	Drop	5	InnoDB	utf8mb4_0900_ai_ci	16.0 KiB	-
□	Report_Problem	☆	Browse	Structure	Search	Insert	Empty	Drop	0	InnoDB	utf8mb4_0900_ai_ci	16.0 KiB	-
□	UserDetails	☆	Browse	Structure	Search	Insert	Empty	Drop	4	MyISAM	utf8mb4_0900_ai_ci	2.3 KiB	-
	6 tables	Sum							26	InnoDB	utf8mb4_0900_ai_ci	82.3 KiB	0 B

Test Plan

Test Case 1

Test case description:

Verify accessibility/availability of the app, download and installation of app

Input:

Search for Bikemee app
Click download and install (Andriod, iOS and window)
Access Bikemee website from firefox, chrome

Condition or function under test:

Available Internet/network connection
Device with internal memory of at least 8GB
Android system above 4.2.2
Mobile browser compatibility

Expected Output:

App should install on mobile with no issue

Output:

Installed successfully

Test Case 2

Test case description:

Verify that the on-screen keyboard appears when the user wants to enter text

Input:

Click on the box to enter First name

Click on login or register

Condition or function under test:

Interface testing

Basic website functionality test

Expected Output:

On-screen keyboard should be displayed

Output:

Screen keyboard display

Test Case 3

Test case description:

Verify that the GPS map is

Input:

User enter current location

User enter destination

Click on start button

Condition or function under test:

GPS testing

Location service

Mobile feature testing

Expected Output:

GPS map should be live with motion

Output:

GPS is actively functioning

Test Case 4

Test case description:

Check response when valid email and password is entered

Input:

Valid Email: test@bikemee.com

Valid Password: testing123

Condition or function under test:

Login page function test

Enter Email Address
Enter Password
Click Sign in

Expected Output:

Login should be successful

Output:

Successful

Test Case 5

Test case description:

Check response when invalid/unmatched email and password is entered

Input:

Email: test@bikemee.com
Invalid Password: testing122

Condition or function under test:

Login page function test
Enter Email Address
Enter unmatched Password
Click Sign in

Expected Output:

Prompt message "password not match email address"

Output:

"password not match email address"

Test Case 6

Test case description:

Admin receive a notification when the number of bike at a dock station is 1

Input:

Two bike are signed out from a dock

Condition or function under test:

Notification testing
Monitoring the business in real live time

Expected Output:

The manager should receive a notification

Output:

Notification received

Project Backlog

User Story	Tasks	Task Owner	Status
He needs a web-based application that will be used both by Mr. Dowling to manage the business and by customers for renting bikes	Create website for manager at Dowlin inc to be able to manage business activities. • Log in Page • Dashboard for Manager	Team	complete
	Create mobile application for customers to rent Bike • Connect App to database • Figure QR Code implementation	Team	complete
	Create System Architecture	Bader	complete
	Create ERD	Emmanuel and Bader	complete
A customer must set up an account with Dowling Inc. He/she must provide a name, address, birthdate, phone number, valid credit card information and email address	Add customer Table to ERD.	Blessing	complete
	Create sign up page with necessary credentials (name, address, phone number, valid credit card information and email address). • UI for the sign up is complete • Front end and Back end implementation in progress	Team	complete
	Add necessary credentials to the database.	Team	complete
The rental rates for a bike are $5 per half hour, $9 per hour for as long as the bike is rented. If a portion of an hour is used at return, the customer will be charged the full hour. If the bike	Show implementation for credit card validation and rental rate charges in UML diagram and Sequence diagram.	Team	complete

is returned to a different dock, the customer will be charged an additional $25. If the bike is not returned, the customer's credit card will be charged $500. If a bike is damaged, the customer will be charged $200 for the damage, depending on the damage. The customer must indicate knowledge and agreement of these requirements before he/she is approved to rent a bike.	The customer should be required to sign this page after the sign up page.	Team	complete
	Show in use case how bike rental service will work based on the specified rules.	Baihan	complete
	Database updates.	Bader	Complete
	Construct terms and condition page.	Team	complete
Mr. Dowling, the manager, should be able to monitor the dock station on real time basis. The manager should be able to track usage and check for valid credit card before bike is unlocked for a customer	create admin portal that monitors dock station activities based on updated data. • Dashboard for Manager	Team	complete
	Database stores dock station activities and monitors bike usage	Team	complete
	Allow manager to be able to flag customers with invalid credit cards on portal by expiration date.	Team	complete
Manager wants weekly report of bike usage and income of each dock	show that weekly usage can be extracted from the database to the portal.	Team	complete
Github	create github repo for team member	Blessing	complete

Sprint Backlog

Backlog for first sprint

Task Name	Task Owner	Status
Final App name	Team Members	completed
Illustrate Use Case Diagram	Baihan and Bader	completed
Create Stories and Scenarios	Emmanuel and Blessing	completed
Create Github Account and add Team members and TA as collaborators	Blessing	completed

Team mates brainstorm about name ideas for the web application	Team Members	completed

Backlog for second Sprint.

Task Name	Task Owner	Status
Create Sequence Diagram	Team Members	inprogress
Illustrate System architecture	Bader	complete
Create UML Model	Baihan	complete
Requirement specification documents	Emmanuel	complete
Create entity relationship diagram	Blessing	complete

Backlog for third sprint

Task Name	Task Owner	Status
Create Database	Bader	Complete
Sketch UI Design for the app	Emmanuel	complete
Figure Out QR Code	Baihan	complete
Create Log in Page for Manager	Blessing/Bader	complete
Start building app with Flutter	Blessing	inprogress
Commit to git	Team members	inprogress
Test Plan	Team members	inprogress

Backlog for fourth

Task Name	Task Owner	Status
Connect App to Database	Blessing	Complete
Complete Mobile Application	Blessing	complete
Complete Admin Dashboard	Emmanuel/Blessing	complete
Complete Admin login	Blessing	complete
Complete Home Page	Bader	complete
Commit to git	Team members	complete
Test Plan	Emmanuel	complete