

Smart KDelivery App

Summary:

Our startup aims to provide a convenient and efficient bike rental experience for city commuters, tourists, and bike owners through a smart app. The service allows users to quickly find, unlock, and rent bikes across the city. Key features include real-time search, QR code scanning for unlock, ride tracking, digital payments, and personalized bike recommendations based on nearby availability. Users can filter bikes by distance, price, and type, and compare options using ratings, condition, and speed limit. The step-by-step process is designed to be fast and reliable, with features like live ride tracking, favorite bike station saving, and easy auto-payment. Bike owners can list their bikes and set rental availability, while the app also offers weekend promotions, free first ride, and refunds for mechanical issues. Users can also communicate with support or report bike damage. The app will send push notifications for ride end alerts, maintenance alerts, and fast re-booking, providing a seamless experience for all users. Overall, our goal is to make bike rental a hassle-free, enjoyable experience that encourages sustainable transportation in urban areas.

How it helps (business terms):

****Low-Level Design (LLD) for Smart Bike Rental App****

****Modules:****

1. ****User Module****: Handles user registration, login, and authentication.
2. ****Bike Module****: Manages bike listings, rental availability, and bike information.
3. ****Rental Module****: Facilitates bike rental, payment, and ride tracking.
4. ****Payment Module****: Handles digital payments and refunds.
5. ****Notification Module****: Sends push notifications for ride end, maintenance alerts, and re-booking.
6. ****Support Module****: Provides support and bike damage reporting.
7. ****Promotion Module****: Offers weekend promotions, free first ride, and discounts.

****Features:****

1. ****Real-time Search****: Users can search for bikes based on distance, price, and type.
2. ****QR Code Scanning****: Users can unlock bikes using QR code scanning.
3. ****Ride Tracking****: Users can track their rides in real-time.
4. ****Digital Payments****: Users can make payments using the app.
5. ****Personalized Bike Recommendations****: Users can receive recommendations based on nearby availability.
6. ****Favorite Bike Station Saving****: Users can save their favorite bike stations.
7. ****Live Ride Tracking****: Users can track their rides in real-time.
8. ****Auto-Payment****: Users can set up auto-payment for future rentals.
9. ****Communication****: Users can communicate with support or report bike damage.

****Data Flow:****

1. ****User Registration****: User provides information (email, password, etc.).
'ç F F fÆðws to User Module.
2. ****Bike Listing****: Bike owner lists their bike with details (price, type, etc.).

'ç F F fÆðws to Bike Module.

3. **Rental Request**: User requests a bike for rental.

'ç F F fÆðws to Rental Module.

4. **Payment Processing**: User makes payment for bike rental.

'ç F F fÆðws to Payment Module.

5. **Ride Tracking**: User tracks their ride in real-time.

'ç F F fÆðws to Ride Tracking Module.

6. **Push Notifications**: App sends push notifications for ride end, maintenance alerts, and re-booking.

'ç F F fÆðws to Notification Module.

Tables:

1. **Users**: stores user information (id, email, password, etc.)

2. **Bikes**: stores bike information (id, type, price, etc.)

3. **Rentals**: stores rental information (id, user_id, bike_id, start_time, end_time, etc.)

4. **Payments**: stores payment information (id, rental_id, payment_method, etc.)

5. **Bike_Damage**: stores bike damage information (id, bike_id, damage_description, etc.)

6. **Ride_Tracking**: stores ride tracking information (id, rental_id, start_time, end_time, etc.)

APIs:

1. **User API**:

'ç &Vv—7FW%÷W6W& creates a new user account.

'ç Æöv-å÷W6W& authenticates a user.

2. **Bike API**:

'ç Æ—7Eö&-°es` : retrieves a list of available bikes.

'ç vWEö&-°e` : retrieves information about a specific bike.

3. **Rental API**:

'ç &W VW7E÷&VçF Æ requests a bike for rental.

'ç VæE÷&VçF Æ ends a rental period.

4. **Payment API**:

'ç &ö6W75÷ yment` : processes a payment for bike rental.

5. **Notification API**:

'ç 6VæEöæ÷F-f-6 F-öæ sends a push notification to a user.

6. **Support API**:

'ç &W ÷ t_bike_damage` : reports bike damage to support.

Logic Flow:

1. **User Registration**:

'ç W6W" &ðvides information and submits registration form.

'ç &Vv—7FW%÷W6W& ' —2 6 ÆÆVB Fò 7&V FR æPw user account.

2. **Bike Listing**:

'ç &-°e owner lists their bike with details.

'ç Æ—7Eö&-°es` API is called to retrieve a list of available bikes.

3. **Rental Request**:

'ç W6W" &W VW7G2 &-°e for rental.

'ç &W VW7E÷&VçF Æ ' —2 6 ÆÆVB Fò 7&V FR æPw rental record.

4. **Payment Processing**:

'ç W6W" Ö °es payment for bike rental.

'ç &ö6W75÷ yment` API is called to process the payment.

5. ****Ride Tracking****:

'ç W6W" G acks their ride in real-time.

'ç 6VæEöæ÷F-f-6 F-öæ ' —2 6 ÆÆVB Fò 6VæB W6, æ÷F-f-6 F-öâ `or ride end.

6. ****Push Notifications****:

'ç 6VæG2 W6, æ÷F-f-6 F-öç2 `or ride end, maintenance alerts, and re-booking.

'ç 6VæEöæ÷F-f-6 F-öæ ' —2 6 ÆÆVB Fò 6VæB W6, æ÷F-f-6 F-öç0.

This low-level design provides a detailed overview of the components, features, data flow, tables, APIs, and logic flow of the Smart Bike Rental App.

Founder:

Name: Student One

Email: student1@gmail.com

Phone: 9876543210

UML Diagram:

