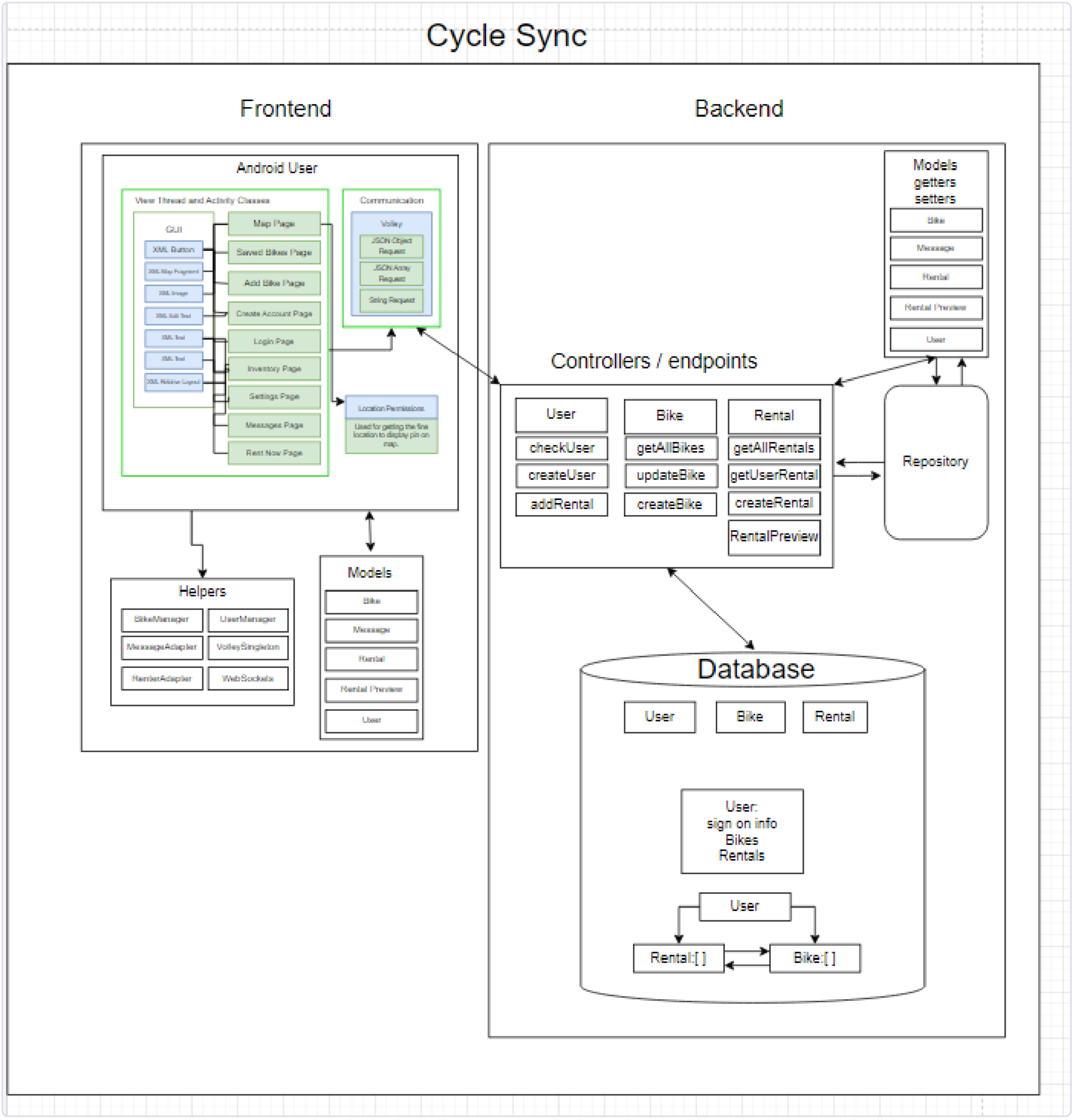
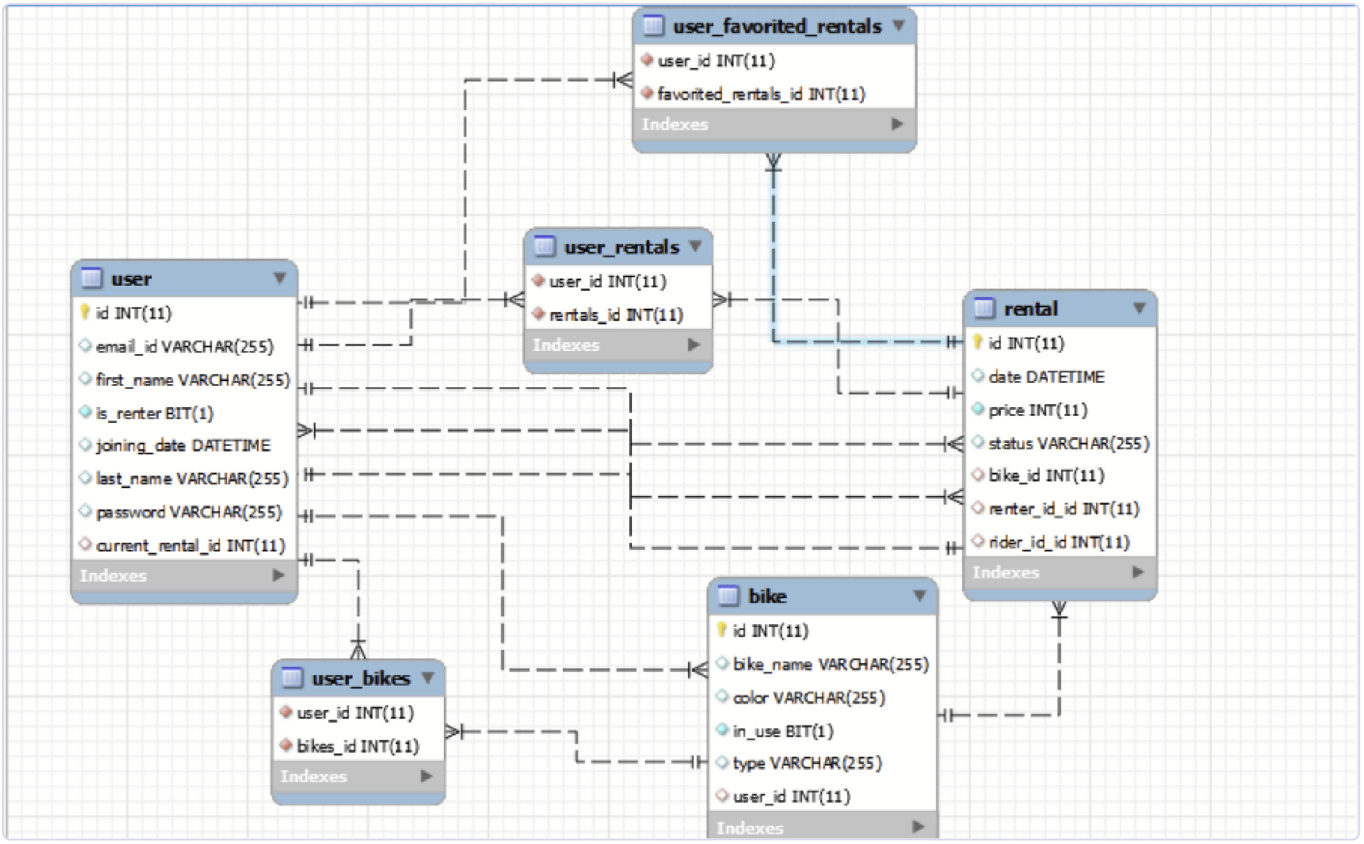
Block Diagram

Team: TA_128
Members: Neil, Thomas, Caleb, Holden
Project Name: CycleSync

Block Diagram



Tables



Design Scription

Frontend

LoginActivity page

LoginActivity manages user login in the app.

UI Elements:
EditText: Username - Input for the user's username.
EditText: Password - Input for the user's password.
Button: Login - Initiates login process.
Button: Create Account - Navigates to account creation screen.

CreateAccountActivity page

CreateAccountActivity facilitates new account creation in the app.

UI Elements:
EditText: Name - Input for the user's full name.
EditText: Email - Input for the user's email.

EditText: Password - Input for the user's chosen password.
Button: Create New Account - Triggers the account creation process.

MapsActivity page

MapsActivity in the app manages the display and interaction of our Google Maps API primarily for locating and selecting bikes to rent.

UI Elements:
Google Map View - Displays the map with bike locations.
Button: Rent Now - Initiates the bike rental process.
Imageview: Settings - Navigates to settings (differing for renter and regular users).

SettingsActivity page

SettingsActivity provides various settings options for the user in the app. It has a different interface for regular users and renters. Renters get an inventory option and that is the main difference.

UI Elements:
TextView: Account Name - Displays the user's name.
ImageView: Back Arrow - Returns to the previous screen.
CustomComponent: Saved Posts - Navigates to SavedPostActivity.
CustomComponent: Become Renter - Leads to RenterSettingsActivity, offering an interface with additional settings for users who are renters.
CustomComponent: Logout Button - Logs out the user and navigates back to LoginActivity.
CustomComponent: Rental History - Opens RentalHistoryActivity to view the user's rental history.

CustomComponent page

CustomComponent is a custom UI element used in the app, designed for displaying various settings options with dynamic content.

UI Elements:
ImageView: Speaker Icon - Displays a custom icon, changeable via attributes.
TextView: Notifications Text - Shows text, customizable through attributes.
RelativeLayout - Used as the main layout to organize content and handle click events.

Designed for flexibility and reuse across different settings.

Data Management

CycleSync's frontend efficiently manages data related to bikes, rentals, and users.

BikeManager: Centralizes management of Bike objects, reducing server requests for bike information. Once a bike's data is fetched, it's accessed and managed locally.
UserManager: Handles user data, ensuring efficient retrieval and updates of user information without frequent server interactions.
RentalManager: Manages rental transactions and statuses, streamlining data handling for rentals.
This structure minimizes server load and network traffic, enhances app responsiveness, and ensures data consistency across the app.

VolleySingleton

VolleySingleton in CycleSync app makes network requests and image loading more efficient.

RequestQueue: Manages all network requests. Singleton pattern ensures a single instance across the app optimizing resource usage.
ImageLoader: Handles image loading and caching. Storing recently used images for quick access.
Context Management: Utilizes application context to prevent leaks in Activities or BroadcastReceivers.
This design enhances network efficiency and reduces memory usage.

Backend

Backend Structure

The CycleSync app's backend is structured with a consistent pattern of Entity, Controller, and Repository for each main class: User, Bike, and Rental.

User:

User (Entity): Represents the user with attributes like name, email, and rentals along with getters and setters.
UserController (Controller): Manages HTTP requests related to users, such as login, creation, and updates.
UserRepository (Repository): Handles data persistence operations for users, interfacing with the database.
Bike:

Follows the same structure with Bike, BikeController, and BikeRepository.
Rental however has RentalPreview which makes it differ more explained below:

RentalPreview

RentalPreview serves as a streamlined representation of rental data in the app.

Purpose: Designed to enhance data efficiency particularly for the RentalHistory screen on the frontend.
Fields: Includes essential details like renter name, bike information (name, type, ID), rental period (start and stop dates), price, and location.
Efficiency: By providing a concise summary of rental information, RentalPreview reduces the need for the frontend to process and display extensive details from the complete Rental entity.
User Experience: Simplifies the display of rental history, presenting users with easily digestible information without overwhelming them with excessive details.

WebSocket Integration

CycleSync uses WebSockets for real-time user interactions within its map and chat features.

WebSocketConfig: Sets up WebSocket endpoints for real-time request handling.
ChatServer (WebSocket Endpoint):Facilitates live chat, managing user connections, messaging, and session tracking.
Key Functions:
onOpen: Registers users, checks for unique usernames.
onMessage: Handles incoming messages, enabling both direct and group chats.
onClose: Manages user disconnections, updates sessions.
onError: Deals with WebSocket communication errors.