**Acceptance Criteria**

**User Perspective**

1. **View and locate the closest bike on a map**
   1. Must display available bikes on an interactive map.
   2. User must be able to see bike locations within an area.
   3. Distance to the nearest bike should be shown.
   4. Map should update in real time to reflect bike availability.
2. **Create an account**
   1. The user must be able to sign up using an email and password.
   2. Must validate the email and evaluate password strength.
   3. Confirmation email after signing up should be implemented.
   4. User should be able to log in using their credentials.
3. **View weather information**
   1. The app must display real time weather updates based on location.
   2. System should fetch data from OpenWeather API.
   3. Weather information must include temperature, humidity, and rain forecast.
4. **View the best route from point A to B**
   1. User must be able to enter a starting and ending location.
   2. System must suggest the most efficient bike route.
   3. The route should consider traffic, bike lanes, etc.
5. **Set location for point A and point B**
   1. User must be able to select a starting and ending location via search or map selection.
   2. System must save the selected locations for future use.
   3. App should allow users to modify their selected locations.
6. **Interactively view the map and see bike station details**
   1. The map should allow zooming and panning for better navigation.
   2. Bike stations should display the number of available bikes and docks.
   3. System must highlight stations with low bike availability.
7. **See different types of information in an easy way**
   1. User must be able to toggle visibility for bike availability and weather.
   2. The app must display all relevant information on a single screen.
   3. The information should be updated based on real time data.
8. **Report a fault or problem with a bike**
   1. The user must be able to report a problem through the app.
   2. System must allow users to describe the issue and submit a photo.
   3. The reported bike must be marked for maintenance and removed from active service.
9. **View how busy certain areas are**
   1. The system must show real time congestion levels in different areas.
   2. App should use bike station data and user density to indicate busy zones.
   3. The information should be colour coded.
10. **See estimated time of arrival (ETA) for locations**
    1. System must calculate an estimated travel time based on distance and speed.
    2. The ETA must be displayed when a user selects a destination.
    3. App should adjust the ETA based on live traffic and weather conditions.

**System Perspective**

1. **Allow users to log in using social media**
   1. System must support Google, Facebook, and other login options.
   2. Users should be able to log in without creating a separate password.
   3. If a social login is used, the app must retrieve the user's name and email.
2. **Handle password recovery**
   1. System must allow users to request a password reset through email.
   2. The reset link should expire after a certain period.
   3. User must enter a new password that meets security requirements.
3. **Store user data for ride history**
   1. System must store ride start and end locations, duration, and distance.
   2. Users must be able to view and filter their ride history.
   3. Data must be stored securely and be accessible only to the user.
4. **Track real time bike location**
   1. System must update bike locations every few seconds.
   2. App must display available bikes.
   3. System must notify users if a selected bike is taken before they arrive.
5. **Get data from OpenWeather**
   1. System must integrate with the OpenWeather API.
   2. Weather data should be updated every 5 minutes.
   3. App must display weather alerts when conditions are unfavourable.
6. **Get data from DublinBikes**
   1. The system must fetch live bike availability from the DublinBikes API.
   2. App must update bike station data in real time.
   3. If the API is down, the app should display a relevant error message.
7. **Get and display user reviews**
   1. Users must be able to submit a review.
   2. System must display the average rating of the bike service.
   3. Reviews must be filtered by date and rating level.
8. **Track bike usage and send alerts for abandoned bikes**
   1. The system must detect if a bike has not moved for an extended period.
   2. If a bike is marked abandoned, an alert must be sent to maintenance.
   3. App should notify users if they forget to dock their bike.
9. **Support different subscription and pricing models**
   1. System must allow users to choose between pay per ride, daily, weekly, or monthly subscriptions.
   2. Users should be able to upgrade or downgrade their plan at any time.
   3. Pricing information must be clearly displayed before purchase.
10. **Set up auto renewal for subscriptions**
    1. Users must be able to enable or disable auto renewal from the app.
    2. The system must notify users before a subscription renews.
    3. If a payment fails, the user must be notified and given a grace period.