SIT725

Systems Integration Overview & Data science SRS

Version 1.0 [Draft]

John Collins
223617689

Contents

Version History

`	Changes by	Update Notes	Comment
1.0	John Collins	Initial Draft	Links need to be added.
1.1			

Documentation Github Links

TBA – link to Document in Github

Project Background

Hi Level Requirements

Design and develop a mobile app solution for finding EV charging stations would have the following features [See Appendix - A]:

1. **User Location:** The app would use the user's location to identify nearby EV charging stations. Users could also enter a destination address to search for charging stations along their route.
2. **Charging Station Map:** The app would display a map showing the locations of nearby charging stations. Users could zoom in on the map to see more details about each station.
3. **Station Information:** The app would provide information about each charging station, including the type of charger, availability, cost, and hours of operation. Users could filter their search results based on these criteria.
4. **Reservation System:** The app would allow users to reserve a charging station in advance, ensuring that they have access to a charger when they need it.
5. **Payment Integration:** The app would integrate with payment systems, allowing users to pay for their charging session directly through the app.
6. **User Reviews and Ratings:** The app would include a user feedback system, where users can rate charging stations and provide feedback on their experience. This would help other users make informed decisions
when choosing a charging station.
7. **Rewards System:** The app could include a rewards system, where users earn points or discounts for using certain charging stations or for using the app frequently.

Product/Solution Vision

The following elaborates on how our product/service distinguishes itself from existing offerings available to users:

Product/Solution Vision	Explanation
For	EV Owners & Drivers
Who	Own or drive electric vehicles that have a limited travel range and need recharging and are faced with limited EV stations/Charge Points and it also restricts EV Owners & Drivers that want to undertake long than EV range journeys.
The product/Service	EVAT Mobile App & Web UI
That	Is a UI/software solution which helps the EV owners & drivers search, find and navigate to EV stations/charge points within Vehicle range, and display this information on a map and provide routing (navigation assistance) to the a desirable EV Charge point.
Unlike	Google maps
Our Product	Provides users, Station Information, with real-time reservation Payment Integration, User Reviews and Ratings, Rewards and additional value-added services (e.g Eco-environmental Information, Station amenities etc.)

System Context

A System Context is a high-level, abstract representation of a software system's interactions with its external entities and boundaries. For this application the external entities are the Enterprise Resource Planning System, Navigation & Location Information Systems, Customer Relationship Management Systems, Reservation (Ordering & Scheduling) Systems and the Payments Integration Systems.

[Diagram to be added]

Solution Implementation: High Level Problem Statement

The target users for the "EVAT" Mobile App and Web Ui would primarily be electric vehicle (EV) owners and drivers. To develop the intended application the following high-level problem statement was developed:

High Level Problem Statement	Explanation
Problem	EV have a limited travel range and need to recharge. There are a growing but limited EV stations/Charge Points.
Affects	EV Owners & Drivers
Which impacts	How far the EV owners can drive before recharging. It also EV Owners & Drivers that want to undertake long than EV range journeys.
A successful solution would be	A software solution which helps the EV owners & drivers, search, find and navigate to desired EV stations/charge points within Vehicle range, displaying location and routing information on a map.

EV User Stories

User stories provide a ‘user-centric’ perspective that helps us understand the user’s needs and expectations. They serve as a guide and valuable tool to the design teams in defining the software’s functionality and guiding its development process.

EV1 Summary Table: User Location (Search & Find EV ChargePoint’s Locations)

Detailed User Story Github link : TBA

User Story	Story Name#EV1: Search & Find EV Chargepoints Locations
As a	EV Owners or Driver.
I Want to	Search, and find a number of EV Charge Stations compatible with my EV (and see real-time availability), in my vicinity (of current or along the route to a planned destination location)
So That	I can select one EV that I can conveniently charge my EV immediately in vacant EV charge points, before my vehicle runs out of charge and without having to wait a long time.

EV2 Summary Table: Charging Station Map with Navigation to EV Charge Point

Detailed User Story Github link : TBA

User Story	Story Name#EV2: Charging Station Map with Navigation to EV Charge Point
As a	EV Owners or Driver.
I Want to	See EV Stations on a Map and Navigate with directions to the desired EV ChargePoint
So That	With the Charging station Map it makes it easy to find and to drive to new EV ChargePoint addresses without getting lost, before my vehicle runs out of charge and without having to wait a long time. Also allows the user to see more details about each station.

Use Cases (Linked to User Story)

User stories are often used to describe the high-level functionality of the system and can be broken down into multiple use cases. Each use case provides a detailed description of how the system will be used to accomplish the goal described in the user story.
The following are the use cases for the Electric vehicle EVAT Mobile & Web App.

UC_1 Summary Table : Search & Find EV ChargePoint's Locations

Detailed User Case Github link : TBA

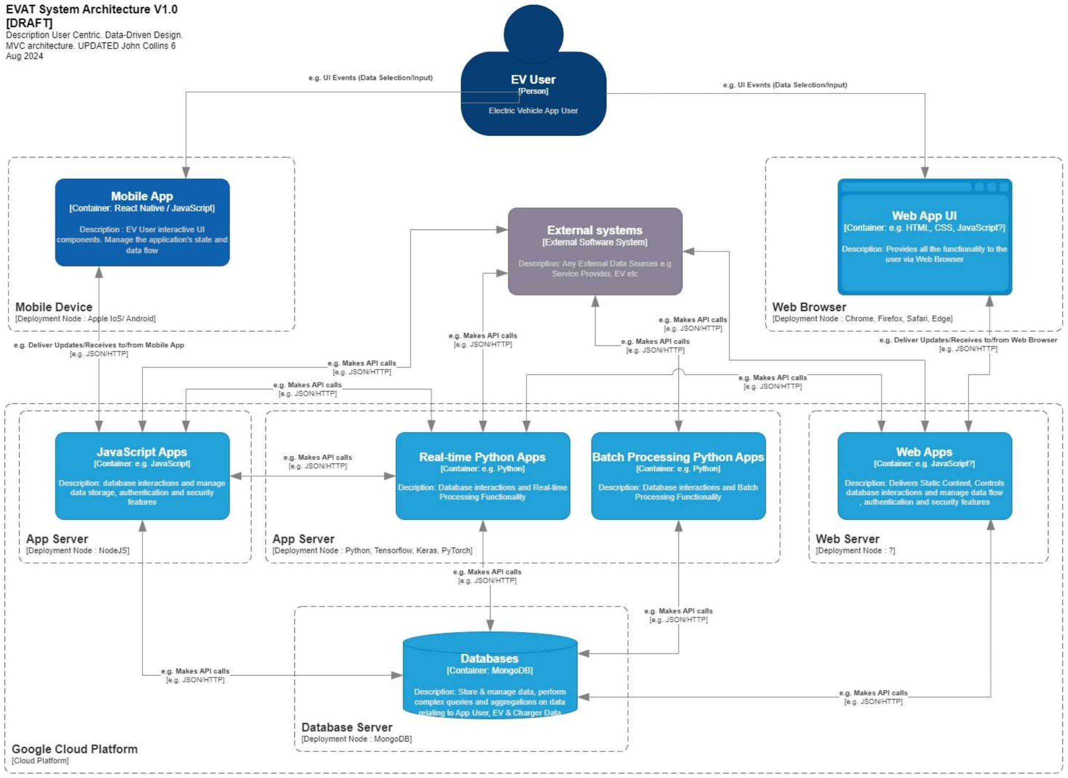
User Case	Use Case# & Name & : UC_1 Search & Find EV Chargepoints Locations
Primary Actor	EV Owners/Drivers.
Secondary Actor	Service Provider
Description	Search & Find EV Chargepoints Locations
Basic Flow (Steps)	• Select current location[Pre-populated GPS] from Drop-down Menu • [Optional] Logged in User, Retrieve User/EV Preferences. Update. • Update Locations of (limited Number) EV Stations on a Map
Alternative Flow1	• Select current location[Pre-populated GPS] from Drop-down Menu • [Optional] Logged in User, Retrieve User/EV Preferences. Update. • [Option] Select destination location[Pre-populated GPS] from Drop-down Menu • Update Locations of (limited Number) EV Stations on a Map along Route
Alternative Flow2	• Enter Current Location, or Obtain Device GPS • Validate Address & Obtain GPS • [Optional] Logged in User, Retrieve User/EV Preferences. Update. • Update Locations of (limited Number) EV Stations on a Map
Alternative Flow3	• Enter Current Location, or Obtain Device GPS • [Option] Enter Destination Location • Validate Address & Obtain GPS • [Optional] Logged in User, Retrieve User/EV Preferences. Update. • Update Locations of (limited Number) EV Stations on a Map along Route

UC_2 Summary Table : EV Station Map with Navigation to EV Charge Point.

User Case	Use Case# & Name & : UC_2 EV Station Map with Navigation to EV Charge Point.
Primary Actor	EV Owners/Drivers.
Secondary Actor	Service Provider.
Description	EV Station Map with Navigation to EV Charge Point.
Basic Flow (Steps)	• Obtain Start location, Destination Location as Per UC_1 • [Optional] Logged in User, Retrieve User/EV Preferences. Update. • Present or obtain EV User Preference Data (e.g operator, amenities) relevant to Journey Planning for selection and filtering. • Present or obtain Electric Vehicle Data (e.g Model, Year, etc) relevant to Journey Planning for selection and filtering • Present or obtain Electric Vehicle Charging Data (e.g Starting %, safety Margin, etc) relevant to Vehicle Station Charging for selection and filtering • Find filtering List of Suitable EV Stations on the planned journey Route. • Select Optimal Stations to complete journey based on previous selections. • Calculate, Distance, Travel and Charging Times. • Display Journey Route on Map, with Charging station stops and Total Distance/Journey time(including charging)

Proposed System Architecture

A high-level overview of a System architecture with a user interfaces (UI), servers, and databases is as follows:

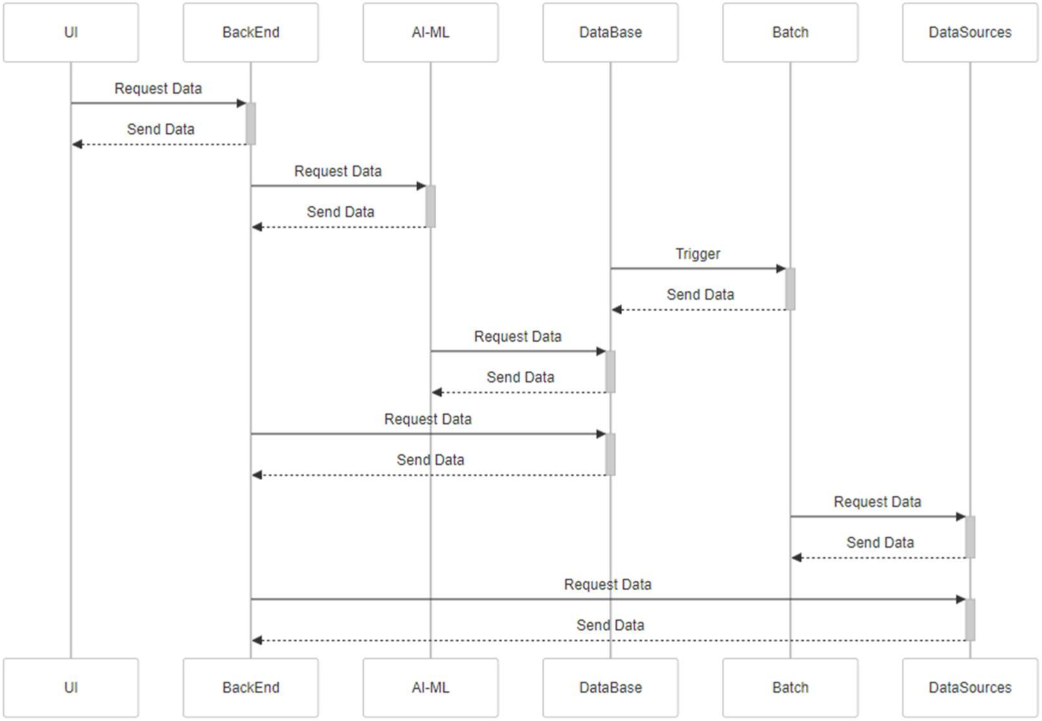

EVAT System Architecture V1.0
[DRAFT]
Description User Centric. Data-Driven Design. MVC architecture. UPDATED John Collins 6 Aug 2024
EV User
[Person]
Electric Vehicle App User
e.g. UI Events (Data Selection/Input)
e.g. UI Events (Data Selection/Input)
Mobile App
[Container: React Native / JavaScript]
Description : EV User interactive UI components. Manage the application's state and data flow
Mobile Device
[Deployment Node : Apple IoS/ Android]
External systems
[External Software System]
Description: Any External Data Sources e.g Service Provider, EV etc
Web App UI
[Container: e.g. HTML, CSS, JavaScript?]
Description: Provides all the functionality to the user via Web Browser
Web Browser
[Deployment Node : Chrome, Firefox, Safari, Edge]
e.g. Deliver Updates/Receives to/from Mobile App
[e.g. JSON/HTTP]
e.g. Makes API calls
[e.g. JSON/HTTP]
e.g. Makes API calls
[e.g. JSON/HTTP]
e.g. Makes API calls
[e.g. JSON/HTTP]
e.g. Makes API calls
[e.g. JSON/HTTP]
e.g. Makes API calls
[e.g. JSON/HTTP]
e.g. Makes API calls
[e.g. JSON/HTTP]
e.g. Deliver Updates/Receives to/from Web Browser
[e.g. JSON/HTTP]
JavaScript Apps
[Container: e.g. JavaScript]
Description: database interactions and manage data storage, authentication and security features
App Server
[Deployment Node : NodeJS]
e.g. Makes API calls
[e.g. JSON/HTTP]
Real-time Python Apps
[Container: e.g. Python]
Decription: Database interactions and Real-time Processing Functionality
Batch Processing Python Apps
[Container: e.g. Python]
Description: Database interactions and Batch Processing Functionality
App Server
[Deployment Node : Python, Tensorflow, Keras, PyTorch]
Web Apps
[Container: e.g. JavaScript?]
Description: Delivers Static Content, Controls database interactions and manage data flow , authentication and security features
Web Server
[Deployment Node : ?]
e.g. Makes API calls
[e.g. JSON/HTTP]
e.g. Makes API calls
[e.g. JSON/HTTP]
Databases
[Container: MongoDB]
Description: Store & manage data, perform complex queries and aggregations on data relating to App User, EV & Charger Data
e.g. Makes API calls
[e.g. JSON/HTTP]
e.g. Makes API calls
[e.g. JSON/HTTP]
Database Server
[Deployment Node : MongoDB]
Google Cloud Platform
[Cloud Platform]

Information Architecture

Hi-Level Data Flow Diagram

A data flow diagram (DFD) is a visual representation of the flow of information through a system or process. It's a fundamental tool in system analysis and design, used to map out how data enters, transforms, and exits a system. Here's a breakdown of interaction between the different system components:

Hi-Level Data Flow Diagram [Draft] V1.0 Updated 9/8/2024 John Collins



User Location: Domain Class Models (Data Entities)

Class Model User Location& Routing Data (Location/Navigation Data/Journey)

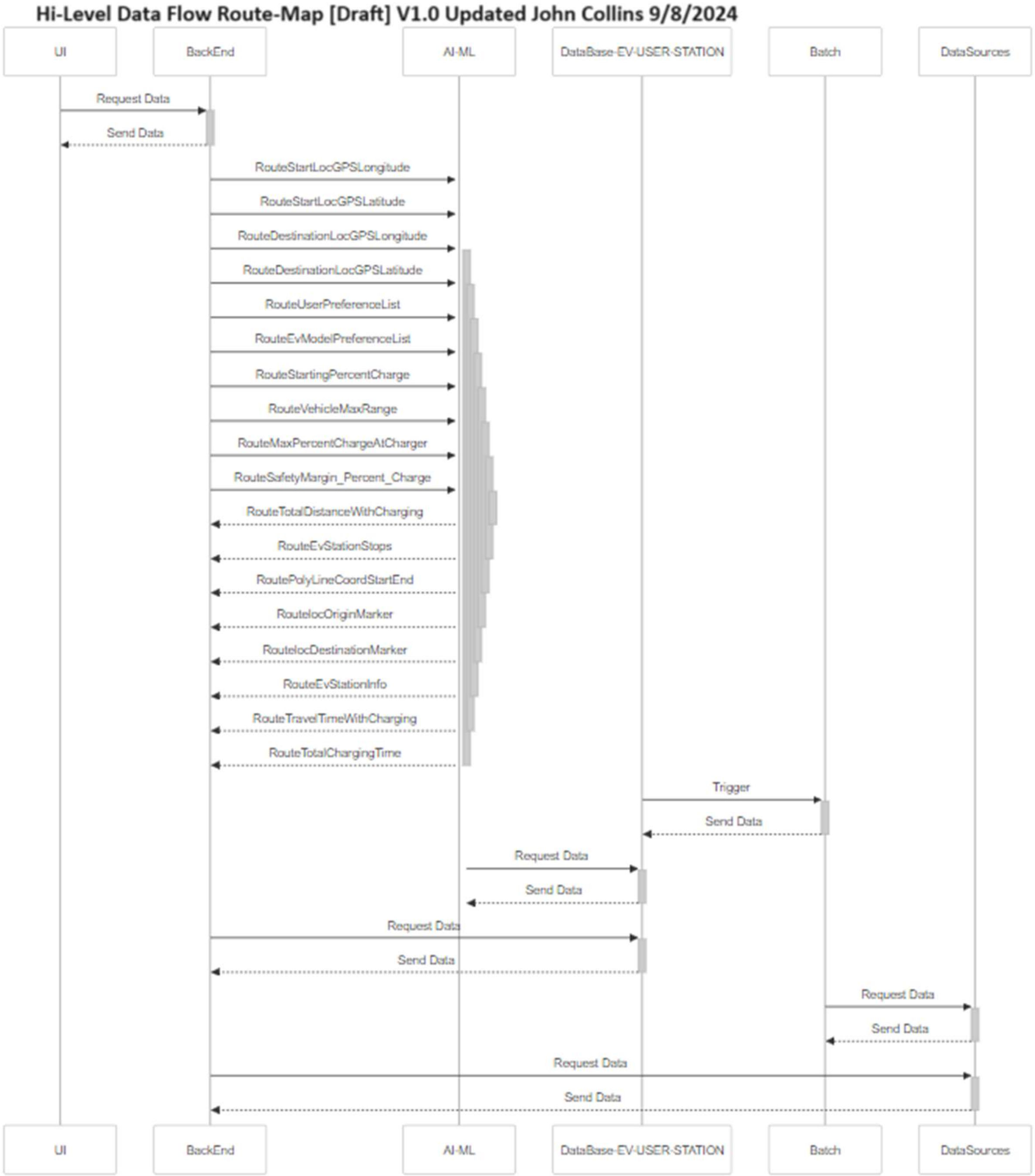
Class Model	Class Model Name: User Location & Routing Data

RouteStartLocMenuList	• Static List Top50 most populated Cities
	• [Optional: Dynamic User Specific List]
	
	`australian_cities_location = {`
	`    "Sydney": [151.2093, -33.8688],`
	`    "Melbourne": [144.9631, -37.8136],`
	`    "Brisbane": [153.0251, -27.4698],`
	`    "Perth": [115.8605, -31.9505],`
	`    "Adelaide": [138.6007, -34.9285],`
	`    "Gold Coast": [153.4000, -28.0167],`
	`    "Canberra": [149.1300, -35.2809],`
	`    "Newcastle": [151.7789, -32.9267],`
	`    "Central Coast": [151.2333, -33.2833],`
	`    "Sunshine Coast": [153.0667, -26.6500],`
	`    "Wollongong": [150.8931, -34.4278],`
	`    "Hobart": [147.3250, -42.8821],`
	`    "Geelong": [144.3500, -38.1500],`
	`    "Townsville": [146.8139, -19.2580],`
	`    "Cairns": [145.7700, -16.9186],`
	`    "Darwin": [130.8456, -12.4634],`
	`    "Toowoomba": [151.9555, -27.5614],`
	`    "Ballarat": [143.8503, -37.5622],`
	`    "Bendigo": [144.2811, -36.7570],`
	`    "Albury-Wodonga": [146.9278, -36.0737],`
	`    "Mackay": [149.1860, -21.1412],`
	`    "Rockhampton": [150.5044, -23.3774],`
	`    "Launceston": [147.1543, -41.4381],`
	`    "Bunbury": [115.6383, -33.3256],`
	`    "Hervey Bay": [152.8400, -25.2888],`
	`    "Maitland": [151.5500, -32.7333],`
	`    "Wagga Wagga": [147.3636, -35.1150],`
	`    "Coffs Harbour": [153.1250, -30.2963],`
	`    "Mildura": [142.1625, -34.2083],`
	`    "Shepparton": [145.3889, -36.3833],`
	`    "Gladstone": [151.2583, -23.8478],`
	`    "Tamworth": [150.9167, -31.0833],`
	`    "Port Macquarie": [152.9185, -31.4333],`
	`    "Orange": [149.1000, -33.2833],`
	`    "Dubbo": [148.6167, -32.2500],`

	"Geraldton": [114.6000, -28.7667], "Nowra": [150.6000, -34.8833], "Bathurst": [149.5765, -33.4193], "Warrnambool": [142.4794, -38.3817], "Lismore": [153.2744, -28.8135], "Albany": [117.8814, -35.0231], "Kalgoorlie-Boulder": [121.4667, -30.7500], "Devonport": [146.3419, -41.1770], "Mount Gambier": [140.7800, -37.8284], "Burnie": [145.9167, -41.0500], "Whyalla": [137.5833, -33.0333] }
RouteDestinationLocMenuList	As per EvUserCurrentLocMenuList above (Minus Start Location)
RouteStartLocAddress	String e.g 100 George Street, Sydney, NSW.
RouteDestinationLocAddress	String e.g 100 Collins Street, Melbourne, VIC
RouteStartLocGPSLongitude	Float e.g 151.2093
RouteStartLocGPSLatitude	Float e.g -33.8688
RouteDestinationLocGPSLongitude	Float e.g 144.9631
RouteDestinationLocGPSLatitude	Float e.g -37.8136
RouteUserPreferenceList	Selectable From: • EvStationAmenities • EvStationOperator
RouteEvModelPreferenceList	Selectable/Prepopulate : Class Model EV User Data • EvMake • EvModel • EvYear • EvMaxRange • EvSocketCompatibility • EvChargePowerOutputs • EvChargeServiceTimes
RouteStartingPercentCharge	Integer e.g
RouteVehicleMaxRange	Integer e.g Range[1 to 400]
RouteMaxPercentChargeAtCharger	Integer e.g Range [1 to 100] (converted to a percentage subsequenctly)
RouteSafetyMargin_Percent_Charge	Integer e.g Range [1 to 10] (converted to a percentage subsequenctly)
RouteTotalDistanceWithCharging	Float (returned Calculation) e.g 659.68
RouteEvStationStops	Integer (returned Calculation) e.g 2
RoutePolyLineCoordStartEnd	Float list e.g PolyLine [(-35.28075, 149.12981),

	(-35.2808, 149.12975), (-35.28087, 149.1297), (-35.28098, 149.12972), (-35.28216, 149.1304) ……. (-34.81646, 148.44058), (-34.81659, 148.43911), (-34.81659, 148.43876)]
RoutelocOriginMarker	Float List e.g [-35.2809, 149.13] GPS
RoutelocDestinationMarker	Float List e.g [-37.8136, 144.9631] GPS
RouteEvStationInfo	String List e.g StationI etc [{'ID': 296044, 'Distance_LastLeg': 282.74, 'Distance_Remaining': 376.85, 'Coordinates': [147.3171930996939, -35.7229067434558], 'Coordinates reversed': [-35.7229067434558, 147.3171930996939]}, {'ID': 210929, 'Distance_LastLeg': 314.57, 'Distance_Remaining': 70.63, 'Coordinates': [144.952084, -37.294362], 'Coordinates reversed': [-37.294362, 144.952084]}])
RouteTravelTimeWithCharging	TBC - estimated
RouteTotalChargingTime	TBC - dervived
Time_stamp	DateTime

DataFlow User Location& Routing Data (Location/Navigation Data/Journey)


Hi-Level Data Flow Route-Map [Draft] V1.0 Updated John Collins 9/8/2024

EV Charger Point : Domain Class Models (Data Entities)

Class Model EV Station Data (Asset Mgt/Location/Navigation Data)

Class Model	Class Model Name: EV ChargePoint Status Data
EvStationId	String e.g 1638722815
EvStationGPSLongitude	Float e.g 144.9631
EvStationGPSLatitude	Float e.g -37.8136
EvStationAddrPostCode	Integer e.g 2000
EvStationAddressLocation	Wollongong, NSW
EvStationGroupId	String e.g 1638722815
EvStationSocketType	String List [See Electric Vehicle Data Model : EvSocketCompatibility e.g Plugs_Type2 Plugs_Three_Phase Plugs_CHAdeMO Plugs_CCS/SAE Plugs_Tesla Plugs_J-1772 Plugs_Caravan_Mains_Socket Plugs_wall_AU/NZ
EvStationPowerOption	String List [See Electric Vehicle Data Model : EvChargePowerOutputs] e.g Rapid, Fast, VAC
EvStationVacant	Boolean e.g True/False
EvStationAmenities	String list e.g {'amenity': 'charging_station', 'brand': 'ChargePoint', 'brand:wikidata': 'Q5176149', 'name': 'ChargePoint', 'operator': 'ChargePoint', 'operator:wikidata': 'Q5176149'}
EvStationOperator	String e.g ChargeFox
EvStationCurrentWaitingTime	?DateTime
EvStationBookedTimeslots	?TBC
Timestamp	DateTime

User Registration & EV Specific : Domain Class Models (Data Entities)

Class Model EV User Data

The Domain Class Model for the user is as follows:

Class Model	Class Model#1: EV User Data
UserFirstName	String e.g Ev
UserSurname	String e.g User
UserHomePostCode	Integer e.g 2000
UserEmailAddress	String e.g evuser @me.com
UserMobilePhoneNumber	String e.g 0400111222
UserAuthenticated	Boolean e.g True/False
UserCurrentVehicleRego	String e.g 1EV2CHARGE
UserAdditionalVehicleRegos	String List String e.g {1EV3CHARGE, 1EV4CHARGE …}
Timestamp	DateTime

Class Model Electric Vehicle Data

The Domain Class Model for the Electric Vehicle is as follows:

Class Model	Class Model: Electric Vehicle Data
EvMake	String e.g Tesla
EvModel	String e.g Model3
EvYear	Integer e.g 2019
EvMaxRange	Integer e.g 400

EvSocketCompatibility	String List e.g Plugs_Type2 Plugs_Three_Phase Plugs_CHAdeMO Plugs_CCS/SAE Plugs_Tesla Plugs_J-1772 Plugs_Caravan_Mains_Socket Plugs_wall_AU/NZ
EvChargePowerOutputs	String List e.g Rapid, Fast, VAC
EvChargeServiceTimes	Integer List
EvRego	String e.g 1EV2CHARGE
EvAssociatedMobilePhoneNumbers	String e.g 0400111222
EvAssociatedRegisteredEmailAddresses	String e.g evuser @me.com
Timestamp	DateTime

Data Model

The domain class model has been previously described. The data model focuses on the physical representation of data within the chosen storage system (e.g., relational database, NoSQL database).

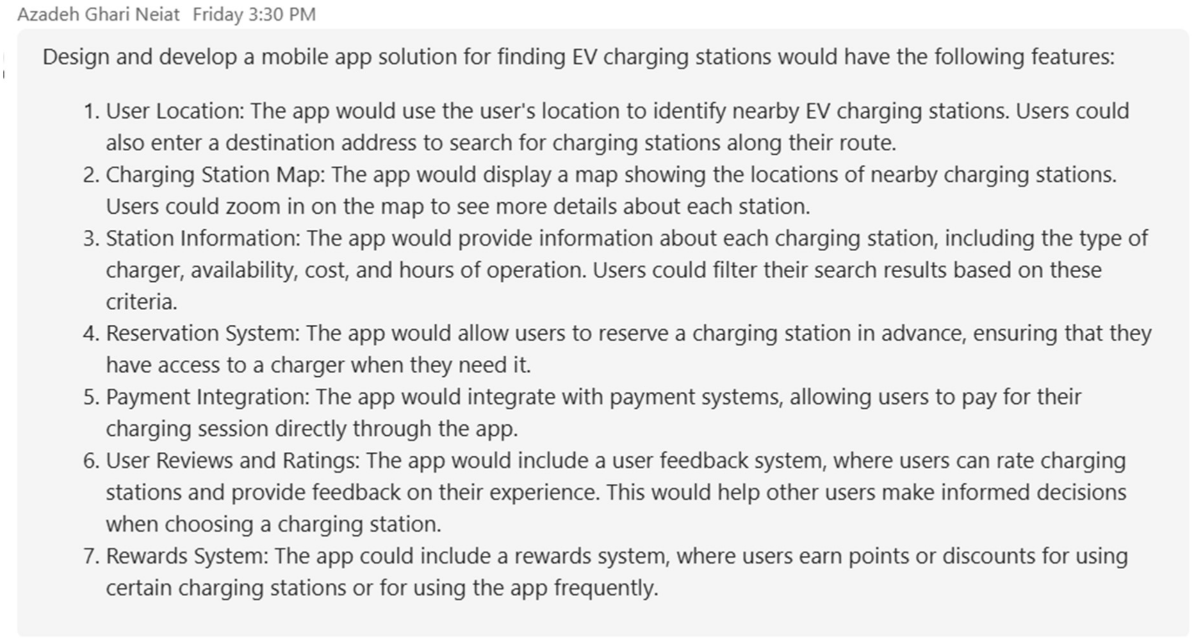
- Defines the structure of data, including tables, columns, data types, constraints, and relationships between tables.

MongoDB Implementation: Bloom Identification Data Model (MongoDb)

TBA

Appendix – Record Decisions/Meeting Minutes

[A] Company Directors meeting 26/7/2024



Azadeh Ghari Neiat Friday 3:30 PM

Design and develop a mobile app solution for finding EV charging stations would have the following features:

1. User Location: The app would use the user's location to identify nearby EV charging stations. Users could also enter a destination address to search for charging stations along their route.
2. Charging Station Map: The app would display a map showing the locations of nearby charging stations. Users could zoom in on the map to see more details about each station.
3. Station Information: The app would provide information about each charging station, including the type of charger, availability, cost, and hours of operation. Users could filter their search results based on these criteria.
4. Reservation System: The app would allow users to reserve a charging station in advance, ensuring that they have access to a charger when they need it.
5. Payment Integration: The app would integrate with payment systems, allowing users to pay for their charging session directly through the app.
6. User Reviews and Ratings: The app would include a user feedback system, where users can rate charging stations and provide feedback on their experience. This would help other users make informed decisions when choosing a charging station.
7. Rewards System: The app could include a rewards system, where users earn points or discounts for using certain charging stations or for using the app frequently.

Background Information : Data Science Jupyter Notebooks Variables & Github links

EV_Route_Planning.ipynb

Jupyter Notebook Name	Type	Shape	Example Value	Desription
Starting_Percent_Charge	int		95	This is the current charge percent of the vehicle at the start of the journey.
Vehicle_Max_Range	int		400	This is the range of the vehicle when charged to 100%
Max_Percent_Charge_at_Charger	int		80	This is the maximum amount of charge that will be applied when stopping at an EV Charger. When stopping and charging at charging stations the rate of charge (speed) is dependent on the current charge amount. Charging vehicles from 80% to 100% is typically much slower than the rate of charge from 10% to 80%. Therefore when travelling it is unlikely you will charge to 100% unless stopped overnight.
Safety_Margin_Percent_Charge	int		10	This is the amount of charge the user would like remaining when they reach a charging station. Think of this like the petrol light. The algorithm will limit the

				driving distance such that the vehicle charge percent will not go below the safety margin
Location_Type_Selector	bool		TRUE	Location Type Selector. Yes = Select from City List. No = Select Long and Lat
Location_Start_City	str	8 chars	'Canberra'	Select your destination from the list of available cities/towns (top 50 by population)
Location_End_City	str	9 chars	'Melbourne'	Select your destination from the list of available cities/towns (top 50 by population)
Longtitude_Start	float		145.1149	Enter you origin (long/lat) and destination (long/lat).
Latitude_Start	float		-37.8475	Enter you origin (long/lat) and destination (long/lat).
Longtitude_End	float		141.4608	Enter you origin (long/lat) and destination (long/lat).
Latitude_End	float		-31.9596	Enter you origin (long/lat) and destination (long/lat).
user_startingpercent	float		0.95	= Starting_Percent_Charge / 100.0
user_safetymargin	float		0.1	= Safety_Margin_Percent_Charge / 100.0
user_maxchargepercent	float		0.8	= Max_Percent_Charge_at_Charger / 100.0
const_maxstops	int		20	The route planning algorithm can get lost in some scenarios. Therefore a limit of 20 stops has been implemented. Testing shows that for the default setup on the current EV charger network it takes 14 stops to navigate from Sydney to Perth. Therefore is the system reaches 20 stops it is assumed that the algorithm has failed to find a route and will exit the loop
op_startingrange				= (user_maxrange * user_startingpercent) - (user_maxrange * user_safetymargin)
op_normalrange				= user_maxrange * user_maxchargepercent
initial_distance_nochargers				RouteTotalDistanceWithCharging
i,				RouteEvStationStops
route_coordinates				RoutePolyLineCoordStartEnd
loc_origin_marker				RoutelocOriginMarker
loc_destination_marker				RoutelocDestinationMarker
df_routeinfo				RouteEvStationInfo
				RouteTravelTimeWithCharging

				RouteTotalChargingTime

References

[1]