Functionality & Context of the Application

Target User

The traffic monitoring app targets general public users who are residing near Minneapolis and St Paul, also including environmentalist-policy makers that intend to use to app to guide their policies and research, and logistics services or delivery couriers who frequently pass through Interstate 94 Westbound and relies on quick and accurate insights about traffic volumes. Users can gain a basic understanding or intuition about the tendencies and patterns between traffic volumes and factors that impact its volume. Knowing the overall variations in traffic volumes and the factors that impact the variations, they can guess and foresee what traffic volume they should expect next thursday, or if it's sunny tomorrow, or on average at 7pm on valentine's day. Although these will not be accurate forecasts based on ML models, it will create a platform in which users can apply filters to the data so that it more or less mimics their unique circumstances and make informed decisions about their commuting habits.

User Capabilities

The system encourages users to read and gain an understanding about past traffic information for MN DoT ATR station with respect to other descriptive features such as the date, time or the applicable weather conditions of the recorded traffic volumes. This way they can customize or personalize their search to mimic circumstances more applicable to their unique scenarios. Past traffic information about this station is really important for analysis and research purposes as it yields historic seasonal or annual patterns accounting for a number of factors that may impact the fluctuations in traffic volumes.

Observations are recorded between the years 2012 to 2018. As this information depicts past traffic instances, most likely users are interested in overall traffic volumes rather than individual or single instances of observations. The reason is, if the users wish to get insights about the tendencies of how traffic volumes behave with respect to specific attributes, they will not be able to get much insight from learning about a single instance. This is because it's likely that a single instance isn't representative of the overall observations, and if they wish to use an individual traffic instance to get insights about their current situations, it's likely that it would not be reliable or accurate. Therefore users are able to perform simple summary statistics limited to 'average', 'minimum' and 'maximum' only on the target attribute; traffic volume. However, they can filter or restrict their search by choosing a filter from the 6 options provided in the dropdown menu

Also stemming from the fact that the data is relatively out-dated or old, users are not allowed to add new traffic instances to the data. First reason is that, if they do so, then there will be a period of at least 5 years between the recently added records and prior records. Another reason is that, it will likely create an issue in terms of the reliability of the data as the observations are limited to a specific station; MN DoT ATR station 301 on Interstate 94

Westbound, accounting for all of the specified attributes that impact the volumes. This is very challenging to capture accurately as an individual.

Hence the system is designed such that users can access basic summary statistics of traffic volumes by querying on attributes that impact the volumes. In the traffic search page, users are prompted to specify a summary statistic to apply on traffic volumes by clicking on one of the options from, average, minimum and maximum. then they make a query or filter from the provided 6 attributes from the dropdown menu, optionally they can make another query to get a more specific result that is more applicable to their current situation.

Although the summary statistics can only be performed on traffic volume, they can be performed based on several filters or queries on select attributes to get a calculation more relevant to a specific circumstance. Users can choose any combination of parameters from the dropdown menu, the query class has all of the 6 parameters that could be initialized with any of them.

The query attributes include date time features such as; year, month and hour (whom for all integer input should be accepted only). They can also be filtered on holiday features such as valentine's day and customized attributes; categorized hour which breaks down a day in textual descriptions i.e. afternoon, and categorized weekday describing the day of the week i.e. monday. For the last 3 attributes users are able to type alphanumeric keywords in string format and the algorithm will match them with corresponding entries. For example, an input of 'noon' should be understood and treated as 'afternoon'. For the textual inputs case sensitivity should also not matter, meaning for example 'thursday' should be accepted and treated as 'Thursday'. Through this query functionality users can access traffic volumes under specific circumstances, for example a user from the general public might want to learn about the average traffic volume on Christmas Day. Or a courier might want to know which hour on monday's yield the least intense traffic volume in order to plan his weekly schedule, etc…

Also in their profile, users can save and track their observations in the 'Observations' page. They can later modify and plot their observations to get a better understanding visually. This functionality allows users such as policy makers to keep track of their observations. Automating this process ensures practicability and easy access to records that may guide their future research and analysis.

Authentication

The system should require control access to API, meaning authentication is necessary. In order to use the services, users are required to register. Process of verifying the identity of individuals can be essential and very convenient for users. This is done so by either creating an account by filling up the registration form stating their name, surname, email address and password or signing up via integrated social platforms. Certain users such as delivery couriers may want to use the app frequently and practically, hence to capture their needs of convenience, the option to sign-up via social platforms such as GitHub and Google is presented in the homepage. User credentials are retrieved from and matched with the credentials from the chosen integrated social platform. By doing so, users avoid the

mandatory steps required for registering an account form, and they can quickly sign-up using their GitHub or Google credentials.

After registering, during the login process, if the user enters an incorrect password for the first time, the page is refreshed displaying the error message prompting them to re-enter the password. If an incorrect password is entered again, users are given the option to reset their password.

Users such as researchers or environmental policy makers and even the general public may want to keep track of their observations, so that once the observations pile up later on, they can perform analyses on them to discover patterns and trends between traffic volumes and their relationships with chosen attributes. Understanding which attributes have the most impact on traffic volume, and investigating reasons which contribute to major fluctuations/variations, can provide a basis for researchers to support and refine their findings. Hence, keeping these observations secure by authenticating access to the API, gives an assurance for users that their personal information and observations are private only to themselves.

Users are able to modify, add, or delete fields of their profile which include their account details and observations page. As these fields are authenticated such that only the users themselves can access and edit and no user can actually change or add new traffic volume records, unauthorized changes that could possible hinder or impact the reliability of the data is not allowed.

Layout of the Application

Accounting for the fact that the user segment will want to use the app on-to-go, practical, and quick, the screen size of the app will be designed for a user audience primarily using a phone, this implies using a layout that most likely fits a portrait landscape. The functionalities within the system and the overall usage of the application should be very easy to use to satisfy the practicality and convenience needs of the target users. Acceptable response time to perform an action therefore shouldn't be long as the users desire the app to be fast, and ready-to-use.

Peak Times of Use

At this stage of the project the app does not yet deploy a ML model (prediction), however it is a functionality that is intended to be corporated into the project at a later stage. Although with the prediction functionally, peak times of use will be more clearly defined, more or less similar assumptions hold at this stage. Conditions which yield the highest traffic volumes are expected to be peak times of use. In more detail, the peak times of use could be expected to be exceptionally high at early mornings and afternoons. This implication is supported by the visualizations of the relationships between traffic volume and categorized hours explored in data preparation steps where it has been indicated that traffic volumes were very high at times where the majority of people go to and leave from work or school. These are expected

to be peak times of use as the majority of users are expected to require traffic forecasts mostly during these times of the day where they are in a rush to get to their destination in time. Same implication holds at this stage of the project as users are likely to be using the application to get some understanding into what traffic volume to expect given their specific conditions or circumstances. So they might simply want to know more or less, on average, what traffic volume was recorded within the 6 year time frame of observations at a specific hour for instance so that they can generalize this implication and use it to plan their time more efficiently.

REQUIREMENTS

Clarification: Different terminologies such as environmental policy makers, researchers, delivery couriers, etc. are used to describe the same target audience that is the general public. Although ultimately they possibly have different reasons for using the app, they are restricted to the same conditions, meaning they all can only modify their own personal details in the account, access and read tailored traffic observations and save their observations in their profile, and none can actually add, update, delete or make any sort of alterations to the actual traffic data.

Requirements Elicitation

The chosen technique for eliciting and initially identifying the requirements is brainstorming the different scenarios under which people will use the app and their intentions of using the app for. From BABOK's elicitation techniques, through brainstorming and review of similar app functionalities I have gained an understanding of the work that the stakeholders do and how the system could support that work. As new ideas are introduced, they're written down so that what is being captured can easily be seen.

Requirements Specification and Documentation

To specify and document the requirements, they have been formalized implementing user stories to capture overarching user needs and also non-functional requirements. Use cases are also implemented to provide additional useful details that may have been missed in user stories.

User Stories

In agile methods, user stories are a great way of capturing example scenarios as they depict all possible routes the user may take, also highlighting alternative routes and aims to

consider all possible usages of the app. Hence, the requirements captured in brainstorming through example scenarios are documented in the form of user stories.
As requirements need to be tracked throughout the project, and possibly link them to other activities such as testing at later stages, they are uniquely referenced.

The requirements are documented initially using agile user stories to discover and reveal certain aspects of the product functionality from the end user's perspectives. Users stories highlight certain key functionalities easy to see from the user's side but challenging to capture from a developers perspective. In an agile framework, user stories act as the end goal to meet software requirements and are expressed from the customer's perspective. Through effective and easily comprehensible discussions, they focus on varying business perspectives and empathize delivering the highest value to the end-users needs. Also the priority is always the customer requirements, this contributes to risk aversion by eliminating possible financial, technical, commercial or communication-related risks. (Benefits of writing good user stories in the agile framework, no date)

Use Cases

As an additional technique to document the requirement and add value, taking the user stories as base, use cases are also constructed. Use cases are also implemented as, in comparison to user stories, the sequence of steps required to accomplish that particular task is outlined in more detail. Use cases are written descriptions of how users will perform tasks within the application. From the user's point of view they outline how the system should behave as it responds to a request. Most importantly, by capturing alternative flows they also reveal what could go wrong within the process. This creates the grounds for approaching solutions to possible problems that may occur before they actually start causing issues. Each use case elaborates a set of goals that can be used to establish the cost and complexity of the system. (Use cases, no date)

Requirements Prioritization

MoSCoW Prioritization method

To prioritize the requirements and the customer needs, I used the MoSCoW method to determine and rank the importance of requirements. With clear prioritization at each stage of the system, MoSCoW technique allows for agile production in stages. As there are no strict time limits, Agility for scheduling and implementation is established and as progress is made, priorities can be refined and adjusted by updating the task list accordingly.

In comparison to other methods such as Forced pair comparison or 100-point method, rather than giving a precedence of priorities based on numerical rankings, MoSCoW method explicitly defines the ranking in words, focusing priorities on decisions that bring value. The MoSCoW method avoids the additional steps necessary in alternative methods that require interpreting the significance of the numerical orderings. Instead, in actual words and in a tangible and explicit way, it gives clarity about where to invest efforts and why. Also, scheduling with MoSCoW is fast and transparent as it does not require a deep

understanding or complicated calculations. Hence MoSCoW model is relatively more simple and straightforward from the developers perspective as to which initiatives to tackle first and which could be postponed to later.

A disadvantage of MoSCoW is that it may create imbalances between the required and slightly desirable requirements. However, although there may be certain ‘floating tasks’ that cause blurred lines and are hard to decide where to place i.e. Must and Should have, considering the scope of this project and the fact that there are relatively less technical limitations and dependencies, it is very straightforward and effective to use. (The most popular prioritization techniques and methods: Moscow, Rice, kano model, walking skeleton, and others, 2019)

User stories

Ref	**User story**	**Acceptance Criteria / Conditions of Satisfaction**	**Outlined requirement**	**MoSCoW Prioritisation**
US01	As a delivery courier, I want to be able to use the web app on all versions of both my IOS work device and Android personal phone so that it’s compatible/works with most of my devices	1. Allow access to download the app through Apple and Google store 2. Support last 5 versions of IOS and Android 3. Response time should not be more than 3 seconds	Non-functional Requirement ***Indicated functionality:*** Compatibility with recent versions of common operating systems such as IOS and Android	Must have
US02	As a member of general public, I would like login from web browsers such as Firefox and Chrome so that I can still access my account if I don't have my personal phone with me	1. Support latest 5 versions of Firefox and Chrome 2. Compliance with cookie policy 3. Access to API must be controlled /authenticated by making registration mandatory	Non-functional Requirement ***Indicated functionality:*** Compatibility with recent versions of common web browsers such as Firefox and Chrome	Should have
US03	As a member of the general public I want to know on average, what traffic volume I should expect on Fridays so that I won’t be late to pick up my kid from his basketball practice at 6:00PM	1. String values denoting the day of the week must be accepted as query input 2. Based on query input, average traffic volume is calculated and a numeric traffic volume estimate is returned	Functional Requirement ***Indicated functionality:*** Basic summary statistics performed on traffic volumes	Must have

		3. Search input must be limited to one field per search box 4. Optionally, if a second query is requested, the search is performed on another search box with the same specifications, but excluding the prior query field 5. Search language is english 6. Data is retrieved from interstate traffic repository using an authenticated API	based on query input(s)	
US04	As a environmental policy maker, I want to observe the average impact of different hours on traffic volume so that I don't spend unnecessary time manually performing every calculation myself	1. Numeric inputs denoting hours of the day are accepted 2. Query box input will depend on the accepted input format of the chosen query field, i.e. query box must only accept numerics if chosen query parameter requires numeric input 3. Based on query input, average traffic volume is calculated and a numeric traffic volume estimate is returned 4. Search is limited to one field per query box 5. Search language is english 6. Data is retrieved from interstate traffic repository using an authenticated API	Functional Requirement ***Indicated functionality:*** Query input is characterized based on data type -numeric or string-	Must have
US05	As a member of the general public I want to see textual descriptions of how traffic volume is likely to behave with my specifications so that I don’t have to interpret the numerical traffic estimates	1. Result should return a duple consisting of the numerical output of the operations and a textual description giving a relative ordering based on the intensity of traffic volume i.e. result: 1147, Relatively low traffic 2. Traffic volume should be described in an additional string column in the prepared dataset, categorized based on the deviation from the mean. There should be 5 categories: {very low, low, moderate, high, very high}	Functional Requirement ***Indicated functionality:*** Textual description of observations is displayed in results	Should have

		3. Search language is English 4. Data is retrieved from interstate traffic repository using an authenticated API		
US06	As an environmental policy-maker, I want to observe and analyze the average, most and least intense traffic volumes based on weather conditions so that I can align my research and policies with prior historic traffic patterns and information	1. Data is retrieved from interstate traffic repository using an authenticated API 2. Results are returned within no more than 2 seconds 3. String keyword inputs (alphanumeric values) describing weather conditions are accepted 4. Keywords must be matched with weather descriptions, allowing specific re-phrasements of the words, i.e. ‘cloud’ or ‘cloudy’ must be understood and treated as ‘clouds’ 5. Search language is english 5. Based on query input, average, minimum and maximum traffic volumes are calculated and a duple consisting of numerical traffic estimate and a string description of the estimate is returned	Functional Requirement ***Indicated functionality:*** Keyword recognition functionality in query box	Should have
US07	As a environmental policy-maker, I want to create an account so that I can track and modify my observations of traffic estimates and change my contact details if needed	1. Access to API must be controlled/authenticated by directing users to create an account 2. The account must capture the user’s first and last name, password and email address 3. Details must comply with data protection legislation 4. In the profile, ‘My observations’ page must be created to optionally record and track user observations in a text field 5. Data must be securely stored	Functional Requirement ***Indicated functionality:*** Account creation and save & track functionality of observations	Should have
US08	As a researcher, I want to be able to easily visualize the relationships between	1. Data is retrieved from interstate traffic repository using an authenticated API	Functional Requirement ***Indicated***	Could have

	my observations of traffic estimates so that I can easily spot insights about traffic patterns that will ease my workload and guide my policies	2. Results are returned within no more than 2 seconds 3. An observations page which includes plot functionality with respect to select parameters must be created 4. For consistency reasons, all observations to be plotted must be the results of single summary statistic, meaning plot function can plot only average, or only minimum or only maximum and not a combination of these operations 5. Plot function will retrieve the chosen summary statistic and query input(s) per observation and create a single instance per observation in the plot 6. Data must be securely stored	***functionality:*** Plot functionality of observations	
US09	As a member of the general public, I want to sign up via GitHub or Google, so that I don't have to enter lots of information and remember my password for yet another website.	1. Providing the option to sign-up via Google and GitHub, 3rd party social login service is integrated 2. User credentials must be retrieved from the 3rd party platform user signed in from 3. Details must comply with data protection legislation 4. Data must be securely stored	Functional Requirement ***Indicated functionality:*** Option to sign-up via third-party social login services such as GitHub and Google	Won't have (for now)
US10	As a developer, I want to record the purposes that the data is being used for, so that I can develop more detailed usage reports and support scenarios with researchers.	1. While registering, the users must be opted to specify their intention of using the data 2. The intention of data use must be restricted to a binary choice task between 'research purposes' or 'personal purposes' 3. The responses should be stored securely for analysis purposes	Functional Requirement ***Indicated functionality:*** Clarification of user's app usage purposes -research or personal-	Could have

Use cases

Use case 01	**Login**
Brief description	User logs in
Primary actors	General public
Pre-conditions	User has an account or have signed up via Google or GitHub
Main flow	1. User enter login details (email, password) and submits 2. System matches if user details match those registered 3.if users details match the registered details, user logs in successfully, and returns to the traffic search page
Alternative flow	3A. Partial details match. Warn the user to re-enter details telling either email or password is incorrect, refresh login page 3B. No match found. Redirect user to either create an account or sign-up via Google or GitHub

Use case 02	**Logout**
Brief description	User logs out
Primary actors	General public
Pre-conditions	User is logged in
Main flow	1. User clicks log out 2. User is logged out of the system and redirected to the homepage
Alternative flow	

Use case 03	**Sign up via online platforms (Google or GitHub)**
Brief description	User signs-up using their credentials from Google or GitHub (integrate Google and GitHub Sign-in with the web app)
Primary actors	General public, Google, GitHub
Pre-conditions	User has either a Google or GitHub account
Main flow	1. User clicks ‘Sign-In with Google’ or ‘Sign-In with GitHub’ 2. User is redirected to enter their Google/GitHub credentials 3. If credentials match, users logs in and returns to the traffic search page
Alternative flow	3A. No match found. Redirect user to create an account

Note: Use case 03 is itself an alternative flow of use case 04 (create account)

More detailed flow

Google Sign-In (Integrating Google Sign-In into your web app, 2022)

1. authorization credentials are created
2. 'Google Platform Library' is loaded
3. App's client ID is specified
4. Google Sign-In button added
5. User profile information is retrieved

GitHub Sign-In (How to integrate github login in your app, 2022)
Configure GitHub Social Login into Your WordPress

1. A new OAuth application is registered in GitHub account
2. An application is registered by entering the Application Name, Homepage URL and Authorization Callback URL
3. Get GitHub OAuth App ID and secret.
4. Configure GitHub identity provider in AD B2C
5. Use GitHub identity provider in user flow.

Use case 04	**create account**
Brief description	User creates an account
Primary actors	General public
Pre-conditions	
Main flow	1. Complete new account form and submit 2. If form validation passes, create new account 3. Redirect user to login
Alternative flow	2A. If form validation fails, prompt the user to correct specific error(s). If the error re-occurs, direct user to sign-up via GitHub or Google

Use case 05	**modify account details**
Brief description	User modifies their personal/contact details
Primary actors	General public
Pre-conditions	Logged in
Main flow	1. Select account field to edit. 2. Enter modified data and submit changes 3. If form validation passes, update field 4. Redirect user to traffic search page
Alternative flow	3A. If form validation fails, prompt user to correct specific error

Use case 06	**query on traffic volume**
Brief description	User apply a summary statistics and query(s) on traffic volume curated by their preferences
Primary actors	General public
Pre-conditions	Logged in
Main flow	Enter answers to below questions: 1. Select statistical operation to perform on traffic volume from the 3 options: average, minimum and maximum 2. Apply the 1st required query chosen from the 6 options provided in the dropdown menu 3. Optionally apply the 2nd query chosen from the same options provided in the dropdown menu (with the condition that 1st query input ≠ 2nd query input) 4. Click on compute traffic volume
Alternative flow	1A and 2A. If fields are left blank, prompt user to correct specific error

Use case 06	**save observations**
Brief description	User clicks on save observation in the displayed results page
Primary actors	General public
Pre-conditions	Logged in
Main flow	1. After making a query/observation, user clicks on the link to save their observation taking them to observations page 2. Traffic results are retrieved from results page and securely stored 3. User clicks on save changes
Alternative flow	1A. observations page can also be accessed directly through user's profile without making a new observation

Use case 07	**modify observations**
Brief description	User modifies observations in the observations page
Primary actors	General public
Pre-conditions	Logged in, user has at least 1 observation
Main flow	1. User clicks on 'Observations' in their profile 2. Users can add, delete or modify any prior or new observation 3. User clicks on save changes
Alternative flow	1A. users can also access observations page through clicking the link to 'save observations' after making an observation

Use case 08	**plot observations**
Brief description	User plots their observations of traffic estimates
Primary actors	General public
Pre-conditions	Logged in, user has at least 2 observations
Main flow	1. User clicks on ‘Observations’ in their profile 2. User clicks ‘plot observations’ 3. User clicks on save changes
Alternative flow	1A. users can also access observations page through clicking the link to ‘save observations’ after making an observation 2A. If user has none or only a single observation, an error message is displayed prompting them to make at least 2 observations to plot

References

Benefits of writing good user stories in the agile framework (no date) *DailyAgile*. Available at: https://dailyagile.com/benefits-of-good-user-stories/ (Accessed: December 18, 2022).

Use cases (no date) *Usability.gov*. Department of Health and Human Services. Available at: https://www.usability.gov/how-to-and-tools/methods/use-cases.html (Accessed: December 18, 2022).

How to integrate github login in your app (2022) *IoT, Web, Embedded*. Available at: https://www.iotality.com/login-with-github/ (Accessed: December 25, 2022).

Integrating Google Sign-In into your web app (2022) *Google*. Google. Available at: https://developers.google.com/identity/sign-in/web/build-button (Accessed: December 25, 2022).

The most popular prioritization techniques and methods: Moscow, Rice, kano model, walking skeleton, and others (2019) *AltexSoft*. AltexSoft. Available at: https://www.altexsoft.com/blog/business/most-popular-prioritization-techniques-and-methods-moscow-rice-kano-model-walking-skeleton-and-others/ (Accessed: December 19, 2022).