Business Requirements Document - SafeRide

Team: #0000FF Thunder
An Ta
Andy Lee
Colin Creasman
Leon Chen
Orion Tang (Team Leader)

Date Issued:
February 5th, 2022

Version History

Version #	Date	Reason for change
0	9/29/2020	Document created
1	10/6/2021	Initial completed version
2	10/27/2021	Updated formatting errors and requirements for features
3	12/12/2021	Updated general requirements to include core components
4	02/04/2022	Updated to include additional features for new group member (Colin)

1. Overview

1.1 Document Summary

This document is intended to outline the project as a whole, including the functionality and specifications that the final product must meet to satisfy the objectives of our “business”.

Section 1, “Project Summary”, provides an overview of the project to be undertaken as well as scope constraints.

Section 2, “General Requirements”, details the general non-functional requirements that are not specific to any use case. Rather, the final product as a whole will adhere to these general guidelines.

Section 3, “Use Cases”, details use cases for the final product. A brief explanation is provided to explain what each section describes.

This document is subject to be updated as seen fit, to keep requirements up to date with the current design. Core functional requirements of the software, however, shall not be taken away in any edit.

1.2 Project Summary

1.2.1 Objectives

- Deliver a commuting companion web application with route safety as its priority
- Allow users to get route directions and see a detailed overview of the safety risks of said route
- Allow users to utilize their profile in order to save route information for ease of access
- Allow users to report any safety hazards they encounter on their travels for other users to see
- Deliver a functional product to the specifications provided in this document

1.2.2 Background

Safety is a concern when it comes to commuting on bicycles, skateboards, scooters, or their electric counterparts. We call these micro mobility devices to generalize the set of human-powered or electric, lightweight vehicles that operate at speeds typically lower than a motor vehicle on the road. Micro mobility users have little choice when it comes to finding a route from point A to point B, as most routing algorithms will calculate the most convenient route in terms of distance or time, and do not let the user know of possible risks on the route. *SafeRide* shall provide its users with transparency when it comes to route safety. It will provide the user with the safest route, calculated by a risk rating system that takes into account factors like live traffic density, accident history of the

route, availability of bike lanes, etc. and give the user a graphical overview of the risks associated with this route.

1.2.3 Benefits

By providing users with safe routes as well as important knowledge regarding what risks are present on this route, we hope to reduce the likelihood of vehicle-micro mobility accidents. Additionally, we hope that more safety-minded commuting applications will create a larger community of micro mobility users which can contribute to a more eco-friendly society.

1.3 Project Scope

1.3.1 In Scope

- Map support for Long Beach City
- Traffic and road-related safety data for Long Beach City
- Provide support for desktop and mobile Chrome users
- During development process, update our services to support latest stable Chrome releases
- English language support

1.3.2 Out of Scope

- After release / due date, support future Chrome releases
- Map support for any area outside of Long Beach City
- Support for browsers other than Chrome
- Satellite imaging for map view
- Real-time navigation that updates live and reroutes if user goes off route
- Offline support
- Specific markers for nearby points of interest on map (akin to Google maps)
- Accessibility features for disabled persons
- Maintenance for services after project due date (however, longevity will be considered in the design process as if this were a true business product)

2. General Requirements

2.1 Sitewide Non-functional requirements

- SafeRide site will be able to support 100 concurrent users. All functions will operate within their specified constraints at this level of usage. Beyond this amount of users, level of performance specified in this document is not a guarantee
- All SafeRide services will be available to users with at least 95% uptime throughout product lifetime. Downtime will account for maintenance and web server / database issues. Scheduled downtime after initial release shall only occur outside of this range of hours: 6am - 9pm

- User input shall be validated so that invalid inputs do not cause fatal errors. System shall let user know that their input was invalid
- User experience shall be accessible to both desktop and mobile Chrome browser users
- 90% of users shall be able to intuitively use the service without guidance
- User shall have the ability to opt-in/out of data collection. Data shall not be shared with any other parties, and is only intended for internal use. Data collection shall not impact core functionality of SafeRide

2.2 Logging

Function
Logging provides a way to track system-wide events for help in code maintenance and auditing

Requirements
- Log entries will be immutable
- Log entries will be saved to a database
- Log entries will detail the Level (Debug, Information, Warning, Error, Fatal), the timestamp when the log was made, and text detailing the log
- Logging process must not impede with use of system
- Logging process must complete within 5 seconds of invocation
- Logging failure must not affect the system's availability

Precondition
Database is available, accessible, and has storage capacity for new log entries

Postcondition
Logging component writes messages with log level, timestamp, and message to the database. Log entries should not be able to be updated from the backend code.

2.3 Archiving

Function
Archiving provides a way to offload data from the database into a compressed file at a different location.

Requirements
- Archiving will take place at midnight local time on the first of every month
- Log messages over 30 days old will be archived
- Archival process will compress log messages into a central location
- After archived, the log messages will be deleted from the database
- Must complete within 60 seconds of invocation
- Archival failures will not result in system going offline

Pre-Conditions
- Data store must be active and accessible
- Archive destination must have space

Postcondition
Log messages older than 30 days will be compressed into a separate location from the database. If successful, logs will then be deleted from the database.

2.3 User Management

Function
Allows for administration of user accounts in the system.

Requirements
- Operations are made on a database
- Only system admins will be able to view User Management view
- System admin will be able to access and modify all accounts in the system
- Single operations - Create, Update, Delete, Disable, Enable
    - Will be completed within 5 seconds of invocation
- Bulk operation - Supports mixed operation
    - Max of 10k operations per request
    - Requests can be made thru an uploaded file
    - File cannot be greater than 2GB
    - Will be completed within 60 seconds of invocation
- System admin accounts can only be created by other system admin accounts
- Failure of this system cannot cause system to go offline

Pre-Conditions
User must be authenticated, on user management view, and a system admin

Postcondition
- Single operations successfully completed within 5 seconds
- Bulk operations up to 10k operations complete within 60 seconds

3. Use Cases

3.x Use case format

Actors
The things involved in carrying out this use case.

Function
A general description of the function of this use case beyond the name.

Precondition
Things that are assumed to be true in order for the use case to be started/the state of the system before starting the use case.

Trigger
The user desire which starts the use case.

Steps
List of numbered steps to the use case, detailing what the actors do to complete the use case.

Postcondition
State of the system after the use case. Essentially, how the state is different from before starting the use case.

Exception
If there are any exceptions that can happen during the use case steps, list what they are and how they are handled. An exception can be considered anything that does not follow the normal flow of steps.

Non-functional Requirements
Measurable quality attributes. Functional requirements are “what the system does”, non-functional requirements are “how they are done”. Needs to be verifiable and able to be validated.

3.1 Register Account

Actors
User, web host, database

Function
User registers their account with SafeRide so they can use our service. Either enters email and password manually or uses Google login

Precondition
Website and database available, internet connection

Trigger
User wants to create an account to utilize all of the site features

Steps
1. User enters an email and password to sign up

OR

1. User chooses google sign in

Postcondition
User successfully registers for an account with SafeRide. Email and Password will be stored in the database for future verification.

Exception
Email entered has already been used to sign up and has a current account with SafeRide. SafeRide displays a warning message and requests alternative actions (Sign In or Forgot Password).

Non-functional Requirements
- Password should have at least 8 characters up to a maximum of 128 characters. Must contain a special character
- Passwords entered directly into our database (not using google login) will be hashed and salted for added security
- Account information will be logged in the database within 1 second

3.2 Log in

Actors
User, web host, database

Function
User signs in to their account with email and password to access their information and past routes in the application

Precondition
Website and database availability, internet connection, user is not already signed in

Trigger
User wants access to full site capability, including saving routes and viewing route history.

Steps
1. User enters their email and associated password

Postcondition
User is successfully logged in and has access to full map features

Exception
If user enters an invalid email/password combination, let the user know that it is invalid and allow them to try again.

Non-functional Requirements
- For a successful login, user shall be logged in within 1 second

3.3 Search for route and get directions

Actors
User, web host, database, third party map API's

Function
User searches for locations to route between

Precondition
Website and database available, internet connection, user is on the homepage

Trigger
User wants to get a route between points and interacts with a location entry box in the "route box"

Steps
1. User enters a search for "starting point" or "destination"
2. Route box displays location suggestions based on what the user types
3. User selects one of these location suggestions and it is automatically entered as "starting point" or "destination" depending on what field is being entered
4. User does this for both "starting point" and "destination" boxes
5. Third party map API returns possible routes between the specified locations
6. System calculates risk rating for possible routes between these locations

Postcondition
Step-by-step directions for the route (supplied by third-party API) with the lowest risk rating are displayed in the route box. The map view is updated to show the route visually.

Exception
User enters a location that results in no suggestions. In this case, in the suggestions list display a message to the user detailing that there are no valid results for that search and they should try a different one.

If the user enters a location that is out of the initial specified map scope, tell the user that they must enter a location within the city of Long Beach.

Non-functional Requirements
- Suggestions lists updates in real time as the user is typing
- Step-by-step directions and the visual route on the map are shown automatically after valid locations are entered
- Risk rating for a route will take no longer than 3 seconds to calculate
- Start and end are shown on the map with markers
- Route calculation and display should occur asynchronously. User should be able to still interact with the map

3.4 Add waypoint to route

Actors
User, web host, database, third-party map API's

Function
User adds waypoints to their route to add multiple stops

Precondition
Website and database available, internet connection, user is on the homepage and has entered at least two valid locations to route between (i.e. user has successfully completed use case 3.2)

Trigger
User wants to add stops on their route to further customize it

Steps
1. User selects "Add waypoint"
2. Another location search bar is added below the last
3. User enters a search for their desired location
4. Route box displays location suggestions based on what the user types
5. User selects one of these location suggestions and it is automatically entered
6. Third party map API returns possible routes between the specified locations
7. System calculates risk rating for possible routes including all specified stops

Postcondition
Step-by-step directions for route (supplied by third-party map API) with lowest risk rating are updated in the route box to accommodate for the new waypoint. The map view is updated to show the new route visually.

Exception
If the user has already the specified maximum number of waypoints to their route, system does not allow user to add more waypoints

Non-functional Requirements
- Suggestions lists updates in real time as the user is typing
- 4 waypoints maximum can be added
- Users can rearrange the order of their waypoints, starting, and end destination by dragging the entry fields to new positions
- Risk rating for a route will take no longer than 3 seconds to calculate
- Each stop on the map will be shown with a marker

3.5 View route history

Actors
User, web host, database

Function
User views a list of previous routes they have calculated

Precondition
Website and database available, internet connection, user is logged in and on the homepage

Trigger
User wants to know what routes they've searched up in the past

Steps
1. User views their account
2. User views their saved routes

Postcondition
The user's route history is displayed on the screen.

Exception
If the user has no past routes, the history list will display a message saying so instead of past routes.

Non-functional Requirements
- Saves the past 30 calculated routes, to be deleted first in-first out if list is full
- Routes are automatically deleted 30 days after calculation
- Each entry in route history displays start and end locations, number of waypoints, and date calculated
- Route history will be displayed within 1 second of request

3.6 Save a route from route history

Actors
User, web host, database

Function
User saves a previously computed route to their account so they can access it for future use

Precondition
Website and database available, internet connection, user is logged in and on the homepage, user has at least one route in their route history

Trigger
User wants to save a route to their account so they can easily access it and display the route and associated data without having to manually enter all locations

Steps
1. User views their account
2. User views their saved routes

3. User saves a route from the list

Postcondition
Route is saved in the system database and associated with said user. Route is available for use in user's "Saved routes".

Exception
If the operation was unable to be completed due to lack of database availability, tell the user.

If the user has saved the specified maximum number of routes already, let the user know they need to make room by deleting a route.

Non-functional Requirements
- User can save a maximum of 10 routes
- Route will be saved within 1 second of request

3.7 Share a saved route

Actors
User, web host, database

Function
Valid routes can be shared via a generated link from the server. Can be imported into other devices.

Precondition
Website and database available, internet connection, user is logged in and on the homepage, user has at least one saved route

Trigger
User wants to get a shareable link to their route to send to someone

Steps
1. User confirms to generate shared link
2. Server generates link
3. Link can be sent to others via text
4. Link can be imported into their device with information visible

Postcondition
Link
Generated link will have a default, constant expiration time limit.

Exception
Link that is expired will not work, cannot be imported. Routes that are still in progress and have not been saved will not be shareable.

Non-functional Requirements
- Default route expiration time limit will be 60 minutes.

3.8 Delete a saved route

Actors
User, web host, database

Function
Saved routes can be deleted, where associated data with the route will be deleted as well.

Precondition
Website and database available, internet connection, user is logged in and on the homepage, user has at least one saved route

Trigger
User wishes to get rid of a saved route in their saved route list

Steps
1. User views their saved routes
2. User chooses to delete route
3. System asks for confirmation
4. User confirms that they wish to delete route

Postcondition
Route is deleted from their saved routes and from the database. Any shared links to a deleted route will automatically expire.

Exception
If database cannot be accessed for any reason, let the user know the deletion could not be completed.

Non-functional Requirements
- Deletion shall occur within 1 second of the request
- Deletion shall require confirmation on the behalf of the user

3.9 Use a saved route to get directions

Actors
User, web host, database, third-party map API’s

Function
Completed routes can be re-initiated into a route in progress where the route is placed on the active map.

Precondition
Website and database available, internet connection, user is logged in and on the homepage, user has at least one saved route

Trigger
User wishes to view a route that they have saved again in the map view, so they can take that exact same route

Steps
1. User views their saved routes
2. User chooses to get directions for a saved route

OR

1. User receives a link to a route
2. User enters link into browser search bar

Postcondition
User is taken to the map view, and their route is displayed on the map. Note that saved routes do not save risk rating; this is recalculated due to dependency on live data so it may differ every time the user views this route.

Exception
If a route link is expired, still take them to the map view but let them know the link is no longer valid.

In the case that the database cannot be accessed for any reason, let the user know that the route could not be retrieved.

Non-functional Requirements
- User shall be automatically taken back to the map view within 1 second of the request. The aforementioned 3 seconds of route calculation time will still be allowed due to the risk rating needing to be recalculated

3.10 Use map overlays

Actors
User, web host, database, third party map API's

Function
Users have the option to select active map overlays on the list

Precondition
Website and database available, internet connection, user is on the homepage

Trigger
User wishes to view color-coded map overlays which detail either traffic, road condition, accident likelihood, and elevation changes for their route.

Steps
1. User navigates to map display
2. User selects an overlay to toggle

Postcondition
Map updates with color-coded visuals to match user selection. Data is retrieved either from database or third-party map API.

Exception
If a route is not currently displayed on the screen, the overlay buttons should not do anything.

Non-functional Requirements
- Overlays are color-coded based on intensity of the specified metric (red - high severity, yellow - medium, no color - no significant severity)
- Overlay shall be applied to the route asynchronously. User should still be able to interact with the map while an overlay is loading
- User can select only one overlay at a time. Toggling another overlay will disable the current one. Similarly, selecting the current overlay again will disable it
- For a valid route, overlay shall be displayed within 3 seconds

3.11 Report hazards

Actors
User, web host, database, third party map API's

Function
User can report hazards that are listed on the display. There will be a prompt that pops up that will allow users to choose from road closures, objects in the road, idle vehicles on the side of the road, accidents, or areas without bike lanes

Precondition
Website and database available, internet connection, user is logged in and on the homepage

Trigger
When a user feels the need to report a hazard on the street so other users can be informed with the hazard that is reported on their route

Steps
1. User views selection of hazard
2. User selects the hazard they wish to report
3. User finalizes hazard placement and report is submitted

Postcondition
Report of hazard will be updated and show up for any other users who will be traveling on the route which hazard was reported.

Exception
In the case the database could not be accessed, let the user know that the hazard could not be reported at this time.

The user can cancel a hazard placement if they do not wish to report the hazard.

Non-functional Requirements
- A maximum of 99 hazards can be reported on the map
- If the same hazard type is reported within 10 meters of another, these are treated as one hazard
- Hazard labels will show how many people reported the specific hazard
- Hazard icons will be easily identifiable at a quick glance
- Hazard markers will persist an hour after their last report. Oldest hazards will be deleted first if the limit is exceeded
- A hazard should be saved in the database within 1 second of user reporting it

3.12 View safety analysis for a route

Actors
User, web host, database, third-party map API's

Function
User can view a detailed analysis of their route safety, which tells them relevant information so they are aware of the risks of the route. It gives details of the route over time, including traffic density, elevation change, road width, accident likelihood (based on historical traffic data), if there were any recent accidents, and when the commuter has to cross an intersection

Precondition
Website and database available, internet connection, user is on the homepage, and has calculated a valid route

Trigger
User wishes to know more about the route and the risk rating associated with that route

Steps
- User views route analysis
- Database and third party API return data regarding route

Postcondition
A section is displayed on the homepage with route analysis utilizing the returned route data

Exception
If there is insufficient data to provide an adequate analysis for a section, let the user know that there was not enough data to provide a report.

Non-functional Requirements

- Data should be displayed visually, using graphs that are easily identifiable
- Analysis should be generated within 3 seconds of the request
- The route analysis element of the UI should be closeable with a single button interaction so that the user can switch back to the map view easily

3.13 Change password

Actors

User, web host, database

Function

User can change their password

Precondition

Website and database available, internet connection, user is logged in and on the accounts page

Trigger

Users wishes to change their password

Steps

1. User views Account Settings
2. User selects change password
3. User is prompt to fill out “Old Password” and “New Password”
4. Pop up box will inform the user of successful change

Postcondition

New password is updated in database, logs user out of current session, prompts new login

Exception

New Password does not meet password requirements, asks user to reenter new password

If the old password does not match the password in the database, tell the user the password entered was wrong.

Non-functional Requirements

- Passwords must be at least 8 characters long with a maximum length of 128 characters, must contain at least one special character
- Password shall be saved in the database within 1 second of the request
- Password will be salted and hashed for security purposes

3.14 Delete account

Actors
User, web host, database

Function
Website and database available, internet connection, user is logged in and on the accounts page

Precondition
Website and database available, internet connection, user is logged in and on the accounts page

Trigger
User no longer wish to keep their account

Steps
1. User views Account Settings
2. User chooses to delete account
3. User will be asked if they are sure to continue
4. Once “Yes, I’m sure” is clicked, the user will be prompted with a pop up window to confirm their account deletion

Postcondition
Account data will be removed from the database as well with any user data

Exception
If the database could not be accessed for any reason, let the user know that the deletion could not be processed.

Non-functional Requirements
- All data tied to the user account shall be deleted from the database
- Account deletion should occur within 1 second of the request
- It must be obviously conveyed to the user that they cannot recover their account once this action is performed

3.15 Add hazard restrictions to route

Actors
User, web host, database, third party map API's

Function
User can select hazards that they want restricted from their route and the algorithm will attempt to generate a route that avoids them.

Precondition
Website and database available, internet connection, user is on the homepage

Trigger
User wants to get a route between points and interacts with a hazard checkbox in the "route box"

Steps
1. User selects locations for both "starting point" and "destination"
2. Page displays empty checkboxes for each type of hazard - "Road Closures", "Objects in Road", "Idle Vehicles", "Accidents", and "No Bike Lane".
3. User selects one or more of the checkboxes as the hazard(s) they would like to restrict from their route
4. Third party map API attempts to return possible routes that avoid the selected hazard(s)

Postcondition
Step-by-step directions for the route (supplied by third-party API) avoiding the selected hazards with the lowest risk rating are displayed in the route box. The map view is updated to show the route visually.

Exception
Third party map API cannot generate any routes between the selected locations that successfully avoids the selected hazard(s). In this case, in the suggestions list display a message to the user detailing that no routes could be generated that successfully avoid the selected hazards and that they should try again with less restrictions.

Non-functional Requirements
- Hazard restrictions update to being "checked" after the user selects them
- Step-by-step directions and the visual route on the map are shown automatically after valid locations are entered
- Risk rating for a route will take no longer than 3 seconds to calculate
- Start and end are shown on the map with markers
- Route calculation and display should occur asynchronously. User should be able to still interact with the map