Gotcha Scooter Mobileye ARD

Amit Moskovitz, Tom Nisim, Alexander Uzelevesky

Table of content

Introduction

Gotcha is developing an accident prevention system for a micro mobility vehicle.
The company demands that we develop in our final project a network for electric scooters that identifies potholes in the road and pedestrians and warns users of the dangers. The project will be integrated with an application to improve the user experience and an administrative tool for data analysis.

1.1) The Problem Domain:

Gotcha company produces a system to prevent accidents in micro mobility vehicles. The company will provide a solution for all these types of vehicles. The system will give 360-degree comprehensive warnings to the rider about the multitude of dangers. The company will have a user interface for the riders that will be dedicated to each user and allow him to use any tool he chooses. The application will be interactive and allow users various options besides the alert system, such as navigating safe routes, recording and analyzing rides, riding together with groups, and more.

1.2) Context:

Our final project will deal only with electric scooters, and we will build a suitable system for them. Also, the application is a new division in the company, and we will create it - the user experience and the management tools for data management.

1.3) Vision:

Our vision is to prevent road accidents, we will do this through system alerts that help micro mobility riders avoid accidents.

In addition, we will increase the sense of security of the riders on these tools.

1.4) Stakeholders:

1. The investors – Gotcha team

2. The developing team - Fourth year software engineering students

3. Clients – Scooter drivers (The main target audience)

4. Scooter's suppliers (Secondary target audience)

1.5) Software Context:

The system will receive as input video received from a camera mounted on the scooter and will warn by visual / audible means after processing the images and identifying various dangers, in addition, the system will offer the rider the safest route based on the input of starting and destination points and will collect information about the road during the ride.

User Profiles - ACTORS

Human:

1) Assigned Users - Our product clients - The users of our system will be the scooter riders, they will register and connect to the application, through which they will be able to contact the GOTCHA team and will receive from the application the safest bicycle routes for travel. The users should mount a Raspberry Pi and a camera on their scooter and will be alerted about upcoming obstacles in the road.
2) Admin - The Gotcha team. They will play a super user role in the system and will be able to send and receive messages from active users of the system, set prizes for outstanding users and see data about all the users (drives, profiles).
3) "Or Yaruk" Organization.
4) Municipalities.

Systems / software:

5) External maps service - An external service of the system. The system will connect to the external maps service in order to receive from it routes according to a destination point entered by the rider.
6) Server - The server will receive and store information from the Raspberry Pi and from the users.

Hardware:

7) Raspberry Pi - A small single-board computer that will be placed on the scooter and will read data (video) from the Camera and analyze it. After analyzing the video, it will decide whether to alert the scooter driver about upcoming obstacles in the road and it will send data to the Server.
8) Camera - It will be placed on the scooter and will be used to record a video of the road ahead of the scooter. The road video shooting will be the input for the Raspberry Pi that will process and analyze it.

Functional Requirements

ID	Module	Description	Priority	Risk
1	Application	The system will support a **System Initialization** operation that includes defining a system administrator and default data (**Appendix A - Configuration File).**	MH	L
2	Application	The system will allow changing and adding external services without affecting the system's operation.	NTH	L
3	Application	the system will allow registration of new users.	MH	L
4	Server + Admin	The system will manage manually Raspberry Pi serial numbers available in the system, and admin will give one for each new rider.		
5	Application	The system will maintain a profile for each user, which includes Age, Gender, Scooter type and driver's license age.	MH	L
6	Application	The system will allow Assigned users to login.	MH	L
7	Application	The system will allow logged on users to logout.	MH	L
8	Application	The system will allow active users to send messages to the system admins.	NTH	L
9	Application	Users will be able to change / reset their password.	MH	L
10	Application	The system will offer the user the safest bicycle routes for travel (**Appendix B - Hazard Rating)** and the travel times for each route.	NTH	H
11	RP	The system will warn of stationary hazards expected during the route.	MH	H
12	RP	The system will detect riding on the type of riding surface (roadway /sidewalk).	NTH	H
13	RP	The system will automatically collect and save information about the locations of the hazards and the type of hazards and the hazard rating according to the algorithm identified on the route.	MH	H
14	Server	The system will automatically update the characteristics of an obstacle if different information is collected about it during another ride, according to the last ride.	NTH	H
15	Application	Admins will be able to delete & add hazard manually.	NTH	L
16	Server	The system will update the relevant municipality and the 'Or Yaruk' association automatically about obstacles by parameters that will be set in the system configuration (obstacle rating, frequency of alerts).	NTH	L

17	Application	Admins will be allowed to update manually a relevant municipality and the ‘Or Yaruk’ association about obstacles.	NTH	L
18	RP	The system will save **riding data** for each ride.	MH	H
19	Application	The system will allow users to view their personal riding history.	MH	L
20	Application	The system will rate users based on **riding data** (according to Appendix C - user rating and awards)	NTH	H
21	Application	The system will allow users to receive delayed notifications. The system must save the notifications and display them to the user when logging in.	NTH	H
22	Application	The system will allow administrators to view the data of all users (trips, rating, profile).	MH	L
23	Application	The system will allow administrators to **delete** and **edit** users.	MH	L
24	Application	The system will allow administrators to add **awards** to **outstanding** users (Appendix C)	NTH	L
25	Application	The system must allow administrators to publish **advertisements** that will be displayed to the user as an image with a link	NTH	L
26	Application	The system will allow administrators to set system configurations. (Appendix A)	MH	L
27	Application	The system will allow administrators to add and remove administrators.	MH	L
28	Application	The system will allow managers to view hazards in the cities by graphical visual view.	NTH	L
29	Application	The system will provide to users detailed explanations of its activity, successes and failures.	NTH	L
30	Application	The system will save and allow the administrators to observe the system's conduct in a range of dates (Appendix D - system usage data)	NTH	L
31	Application	The system must save an up-to-date state in an external database in a way that allows separation between computational activity and data. The separation allows system state restoration and efficient management. The system will save data about user’s details, rides, hazards and advertisements who are published in the system.	MH	L
32	RP	The system will work independently in disconnected mode and will save trips until contacting the server, in case of override memory, the system will save the last rides.	MH	L

Non-Functional Requirements

ID	Module	Description	Priority	Risk
33	RP	The system will support multiple types of alerts (**Alert Types** - Configuration appendix)	MH	L
34	RP	The system will alert the driver of upcoming danger at least **Minimum Distance to Alert** (Configuration appendix) meters before the danger	MH	H
35	RP	The duration of the alert will be measured in seconds and will be set by the admin.	MH	L
36	Application	The system will suggest **Number of Routes** (Configuration appendix) routes to the user	NTH	H
37	RP	The system will identify and alert about an upcoming danger in at most. 0.5 seconds on average, 1 second at peak times (10% of the time) and 0.1 seconds on a regular activity (90% of the time)	MH	H
38	Application	The system must run regularly if the system administrators haven't disabled it.	MH	H
39	RP	The system must alert the system admins and the drivers in case it malfunctions.	MH	H
40	Application	The system must support 10,000 concurrent connections.	NTH	H
41	Application	The system must allow the registration of up to 50,000 users.	NTH	L
42	Application	The system must NOT save any unencrypted personal details of the system users.	MH	H
43	Application	The system will enforce use of the system according to the appropriate permissions of the various stakeholders.	MH	L
44	Application	The system data will be initialized from a configuration file that can be changed by the system administrators.	MH	L
45	Application	The server will run on a LINUX OS.	MH	H
46	Application	The administrator's application will run on a Windows OS (version 10 or higher).	MH	H
47	Application	The rider application will run on all types of mobile devices.	NTH	H

48	Application + RP	The system must be able to deal with a loss of communication between system components and external services.	MH	H
49	Application +RP	The system must be immune to the unexpected behavior of external services.	MH	H
50	Camera	The Camera must record 15 meters ahead.	MH	H
51	Application	The system must maintain an error and events log.	MH	L
52	Application	Administrators must be able to view the events and error log.	MH	L
53	Application	The language of the application user interface will be English, the system will support address in Hebrew as well.	MH	L
54	Application	The Application will try to enter automatically the origin-address at beginning of new ride.	NTH	L

Implementation constraints & Technology

Correctness Constraints:

1) There should be always at least 1 admin.
2) Every user has a unique email address.
3) Only registered users can use the system.
4) All user's personal data must be encrypted.
5) Users' password must be at least 6 characters long and contain both upper and lowercase letters.
6) Admin should be a registered user.

Technology:

- **Raspberry Pi 4** with 8 GB of memory for development.
- **Raspberry Pi CAMERA MODULE V2** camera that includes 12GB memory, 8 megapixels, with focus capability, 3280*2464 resolution.
- **React Native** for rider application & admin application.
- **Spring Boot application-based Linux** for server.

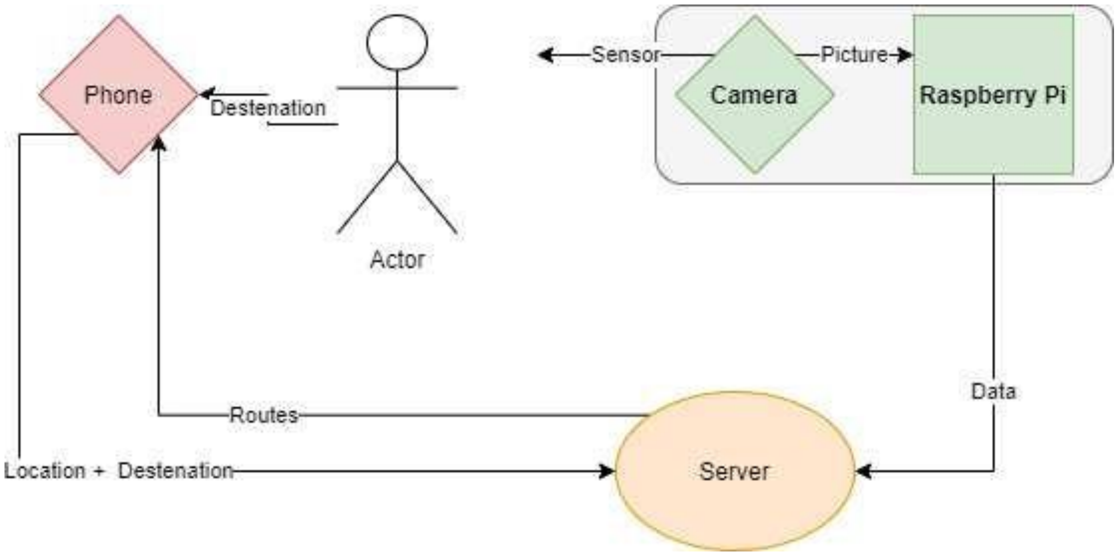
Technology Challenges:

1. Select of RP based on the following requirements: an operating system that supports running Python programs is required, estimate the required amount of memory for the above program and verify enough memory in the device, find out the number of cores required depending on the required response and calculation times in the system.

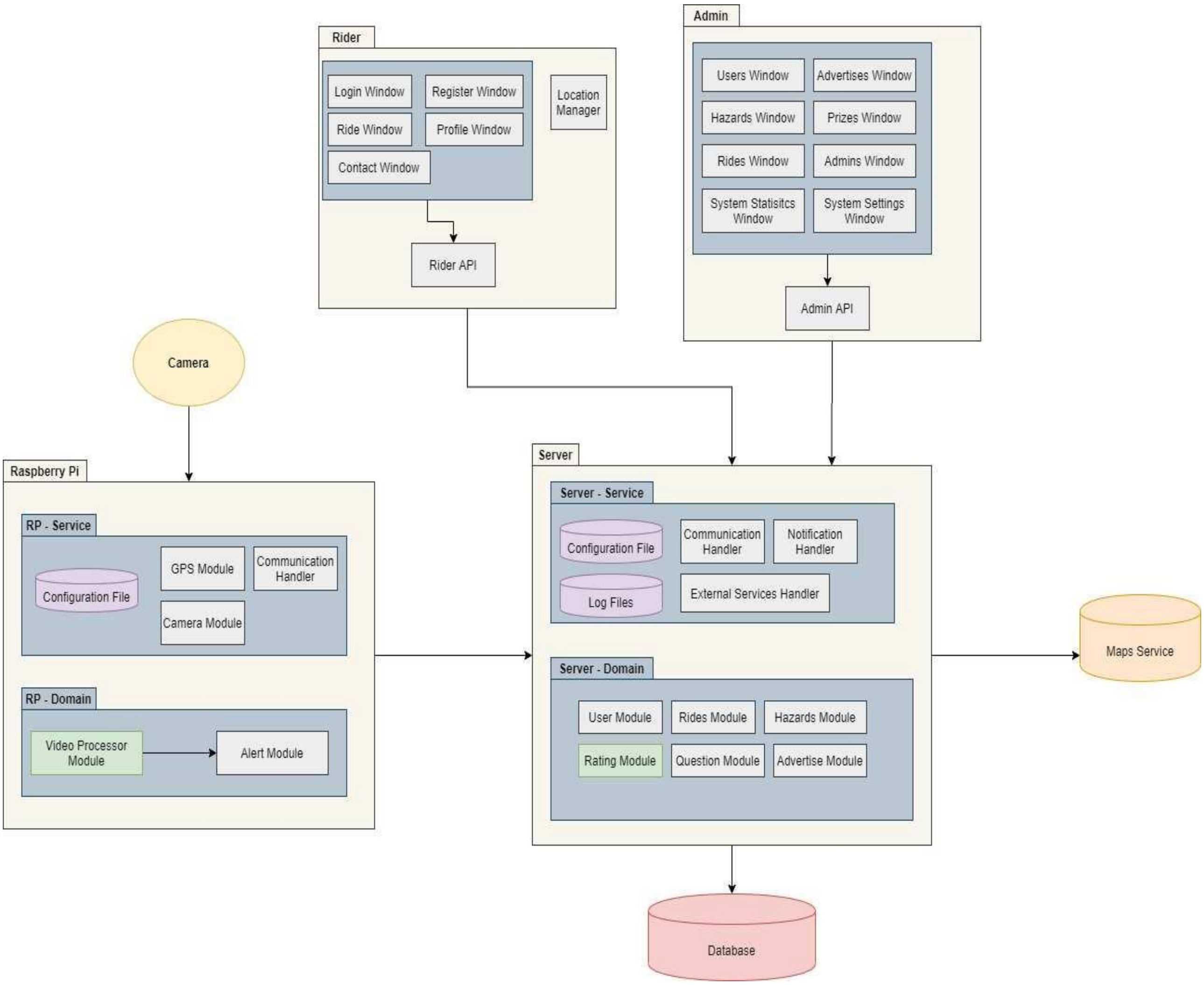
   ** It is necessary to update the requirements according to resources that the system will consume in the expansion of the prototype to the product.*

2. Using external API (Google Maps E.g.) and handling different responses for our requests.
3. Sending data from the RP component to the server - selecting protocol and technology (WIFI), the device is required to connect to a wireless network.
4. Running server on the web - using Docker and cloud services.
5. Exporting rider application program from the REACT NATIVE to the mobile phone operating systems.
6. Choosing & displaying a camera to the RP as input, the camera must shoot 180 degrees, find out what the quality is required from the camera (resolution, etc.)
7. and the difficulties shooting in different environmental conditions - rain, darkness, twilight. The battery of the RP device needs to be charged regularly.
8. The camera will be placed on the scooter and the road might contain a lot of bumps, which may cause the video to be less clear and make the task of recognizing dangers much harder.

System Components



System Architecture Design



Development methodology

General:

The development of the system consists of **3 versions**, where the development approach is characterized by short development cycles.

The versions are based on the General Requirements Document (ARD), this document constitutes the characterization of the system and summarizes the functions of the system.

The role of this document is to help make decisions during the development phase in a way that will satisfy the customer.

Developer tasks:

As a rule, each team member has full responsibility for one of the versions, distributed internally among the team members.

The development process will be conducted in **GitHub** when the work will be conducted in **two-week sprints** that include tasks for each team member.

At the beginning of each sprint, an **Issue** will be opened for each task, which will be tagged by a **team member** responsible for the task**, the number of the related requirement** that is being fulfilled, the **priority** of the task is measured in three types: high, medium and low urgency.

Each developer will open a **branch** for all his personal and document in each commit the changes he made, in each commit code will be added in **small pieces.**

In every feature development, meaningful names for variables, functions, and explanatory documentation for the signatures of the functions in comments will be required, including an explanation of the parameters and the return value, in addition, the number of the realized requirement within the code will be documented.

In addition, in every feature development, **unit tests** are required that cover the possible input space.

Finally, a **code review** will be performed by one of the other team members who will be responsible for merging the branch.

Meetings:

The team will hold **weekly** meetings of 2 types every **Wednesday at 14:00:**

- **Sprint planning** - a meeting that will be held for an **hour and a half** during which tasks will be divided among the team members, the expected difficulties and possible solutions will be discussed. In addition, progress based on requirements will be presented.
- **Cross-section meeting** - a meeting that will take **half an hour** in which we will discuss bugs, interface points between the developers, questions for the client and a progress report.

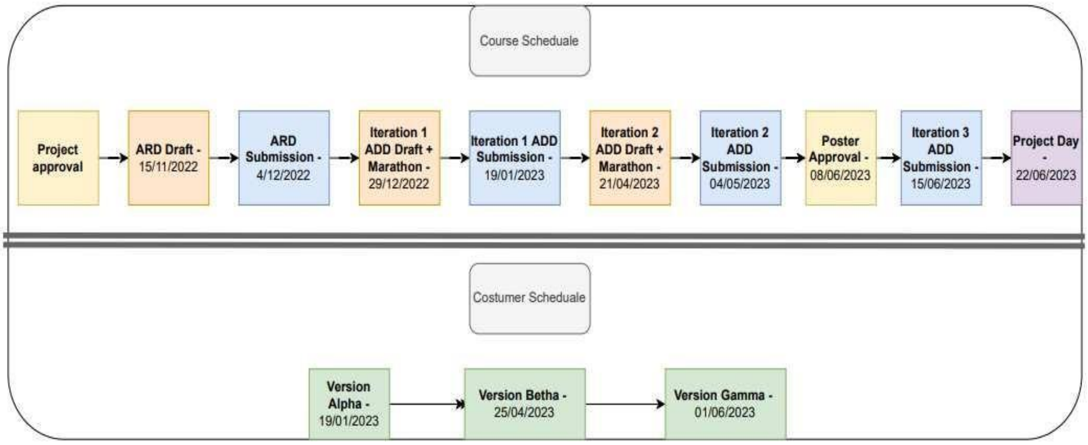
Once a **month** there will be a meeting between the **client and team**, during which the progress will be presented based on the charts, we will present the difficulties and ways of solutions to the client.

Versions documentations:

1) **Glossary -** An alphabetical list of terms consisting of terms that use the system or a part of it.
2) **Usage scenarios** - Scenarios that describe the interactions between the system and the external environment.
3) **ARD -** Updated requirements document.
4) The system development **schedule.**
5) The system **architecture model** (from phase B onward) - a combination of logical and physical components for realization of the system.
6) **Class model** - A document that describes the entities, constraints and relationships that make up the system.

- *Each document will be written in English, 'Calibri' font and font size 11, headline in size 20, background color will be orange for tables.*

Project development roadmap:


Course Scheduale
Project approval
ARD Draft - 15/11/2022
ARD Submission - 4/12/2022
Iteration 1 ADD Draft + Marathon - 29/12/2022
Iteration 1 ADD Submission - 19/01/2023
Iteration 2 ADD Draft + Marathon - 21/04/2023
Iteration 2 ADD Submission - 04/05/2023
Poster Approval - 08/06/2023
Iteration 3 ADD Submission - 15/06/2023
Project Day - 22/06/2023
Costumer Scheduale
Version Alpha - 19/01/2023
Version Betha - 25/04/2023
Version Gamma - 01/06/2023

Alpha version:

In this version, the team members will present a partial implementation of requirements that constitute a significant part of the system that includes a comprehensive testing system that proves compliance with these requirements.

Deadline: 19.01.2023

Manager: Amit

Requirements and metrics:

- Functional requirements - 1, 3, 4, 5, 6,10(partial), 19, 20, 22, 23, 25(partial), 26.
- Non - Functional requirements - 27, 29, 32, 38, 39, 40, 41, 42, 43, 44, 49, 50.
- Initial version of admin GUI.
- Server who supports several operations.
- Photographing and warning of potholes by the camera and the RP.
- Admin application: option to add and remove users.
- Applications are installed on all required work environments.

A central part of the goals of this version is receiving feedback and coordinating expectations with the customer and adjusting the requirements document following the findings (including risk management with the customer's cooperation), in addition, the goal is to provide confidence to customers by running the system on the various hardware components.

Beta version:

In this version, the system will be expanded and the requirements of additional features that complement the system will be realized in it,

In this version, the emphasis will be placed on corrections and adjustments according to the customer's opinion from the alpha version.

Deadline: 25.04.2023

Manager: Tom

Requirements and metrics:

- Functional requirements - 7, 12, 13, 15, 16, 17, 18,21,24, 25.
- Non - Functional requirements - 28, 31, 35, 36, 37, 46.
- Improving the installed program of the camera and RP components according to the requirements.
- Complete implementation of the admin application.
- Realization of the rating part and the selection of the route in the rider application, not yet realization of the presentation of the road during the trip.
- Implementation of a message system between admins and users.
- presentation of statistics and data on system resource consumption - percentages of compliance with requirements and system metrics.

Gamma version:

The final version of the project includes a complete system of tests for all usage scenarios and final completion of all features in the system.

Deadline: 01.06.2023

Manager: Sasha

Requirements and metrics:

- Functional requirements - 2, 8, 9, 11,14.
- Non - Functional requirements - 30,33,34, 45,47.
- The rider app: initial support for the idea of choosing a route and displaying it during the trip.
- Realizing the part of the advertisements and designing the various applications according to the complete requirements.
- A report on faults we have faced and are expected to face in future.
- Data about memory consumption data, running times.

Glossary

ID	Subject	Explanation
1	System Admin	A user who belongs to the development team, can perform operations in the system in order to keep its maintenance.
2	System Initialization	A process that happens when the program is uploaded, includes defining a system administrator and default data.
3	External service	A service that the system connects to, and thus can export some actions like supply and payment to external modules.
4	Mandatory External service	An external service that the system cannot run without.
5	Optional External service	An external service that the system can run without.
6	Assigned user	A user who performed a registration process and thus can perform additional operations in the system.
7	Guest	A user who has not performed a registration process and thus can perform only several operations in the system.
8	Appendix A	A file that defines the configuration settings of the system.
9	Appendix B	A file that defines the manner of hazard ratings.
10	Appendix C	A file that defines the manner of user ratings and prizes awarding.
11	Appendix D	A file that defines the system usage data.
12	User Profile	A collection of information about the user includes Age, Gender, Scooter type and driver's license age.
13	Safest Bicycle Routes	Routes tagged by the system as the safest among the possible routes according to predefined data regarding hazard rating (Appendix B)
14	Stationary Hazards	Stationary objects during the route that can harm safe travel such as potholes in the road, electric poles, etc.
15	Mobility Hazards	Mobility objects during the route that can harm safe travel such as pedestrians.
16	Riding Surface	The surface on which the ride is carried out. Can be a roadway or sidewalk.
17	Hazard location	Define the exact coordinates and the city where the danger is located.
18	Riding Experience Questionnaire	A questionnaire that is presented to the user at the end of the ride and serves as the user's feedback on the ride.
19	Riding Data	Data the system saves and calculates for each ride: average speed, maximum speed, speed change following a warning, amount of braking, number of sharp turns, number of warnings, location of travel on roadway or sidewalk.
20	Riding History	A collection of Riding Data for each ride taken by a specific user.
21	Delayed Notification	Messages sent from admins to users. If the user is connected to the system – he will get the message right away. Otherwise, the message will be displayed when logging in.

22	Outstanding Users	Users who excelled in safe riding and received high ratings for their recent rides.
23	System Explanation	The system will provide a detailed indication of user operations. For example, if a registration operation fails, the system will present an explanation to the user about the reason for the failure, in addition, the system will inform the user when an action he initiated has been completed successfully.
24	System's conduct	Information for the system administrators detailing the usage data of the system such as number of registered users, number of guests, number of rides in a certain time, etc..

Appendix A - Configuration File

1) **Minimum Distance To Alert -** A distance (in meters). The system must alert the driver about an upcoming obstacle when the obstacle is at least this distance away from the scooter.

   Default Value: 5 meters.

2) Alert Duration – how much time the alert will ring? measured in seconds.

   Default Value: 5 seconds.

3) **Alert Types** -
   a) Vibration
   b) Sound
   c) Verbal
   d) Visual

   Default Value: Sound

4) **Number Of Routes** - How many routes will the system show the driver when he chooses his desired destination.

   Default Value: 3

5) **Admin Credentials** - Email and Password of a default admin (To make sure there is always an admin)
6) **Server Address** - IP and Port of the Server to make sure the RP can create a connection to Server in order to send data.
7) **Minimum Password Length** - The minimum length the password must be in order to be acceptable.

Appendix B - Hazard rating

The system will rate the various dangers based on the following parameters:

- Size - the length, width and height will be measured in meters.
- The type of danger - road sign, pothole, pole/tree.

When the rating will be in the range of values **1-20** where 1 signifies the least danger, compared to 20 the maximum.

The sum of the dangers that exist on any road is the road's safety rating, where the lower the rating, the safer the road.

The system will calculate the width of each obstacle in square meter units, and rate the obstacle according to different formulas, so that the rating of the obstacle will be considered in the ratio of the obstacle’s width in a standard path of 2 meters.

Finally, the obstacle rating range will be between 1 and 20 so that a high score indicates a dangerous obstacle.

Obstacle Type	Formula	Range(m)
Pothole	10 + (W/2) *10	10-20
Road Sign	(W/2) *10	0-10
Pole / Tree	5 + (W/2) *10	5-15

Appendix C – Users rating and prizes

The system will calculate an average of the 5 sections below to get the driver grade.

For each drive, the driver will receive coins that will be used to make purchases in the application.

The distributed of the coins:

For each drive, the driver's grade will be calculated according to the instructions above, this grade will be divided by 10 - this value will be the number of coins that rider will receive.

2. Number of speed changes per minute

Speed change: when there is a change of at least 5 km/h.

Grade	Number of speed changes per minute
100	0
95	1
85	2
80	3
70	4
60	5
50	At least 6

3. Number of brakes per minute

Grade	Number of brakes per minute
100	0
95	1
85	2
80	3
70	4
50	At least 5

4. Number of sharp turns per minute

Sharp turn: when there is a change in the direction of driving of at least 60 degrees.

Grade	Number of sharp turns per minute
100	0
85	1
80	2
70	3
50	At least 4

5. Number of alerts per minute

Grade	Number of alerts per minute
100	0
95	1
90	2
80	3
75	4
70	5
50	At least 6

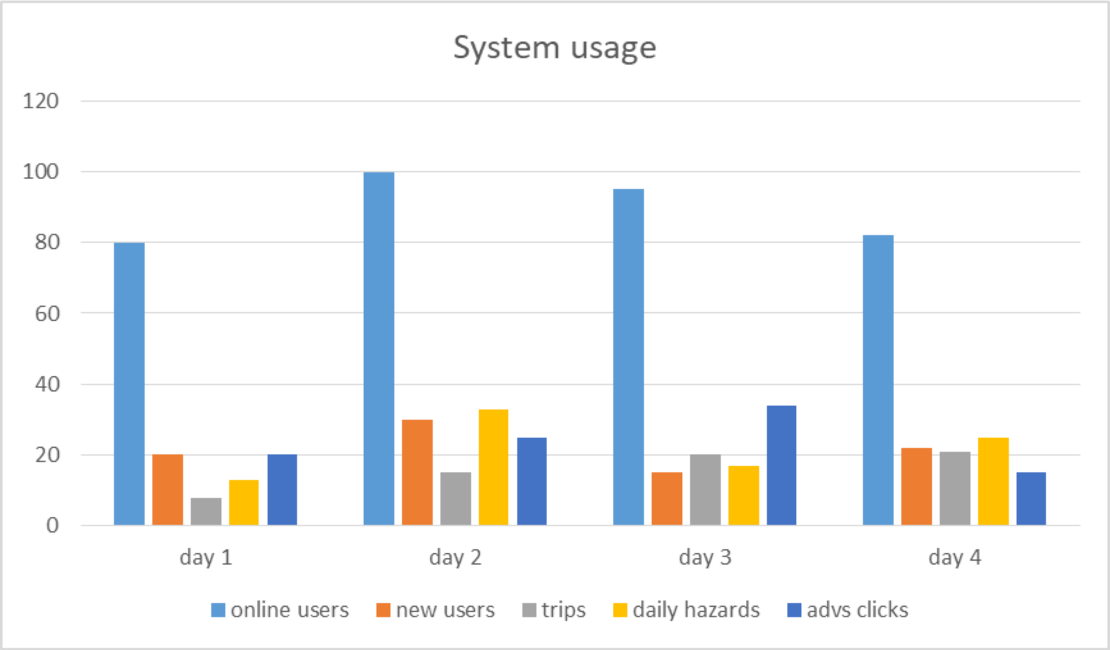
6. Being on the sidewalk (percentage)

Grade	Being on the sidewalk
100	0%
85	10%
75	20%
60	30%
50	At least 40%

Appendix D - System usage data

System administrators can view the following data:

- Number of online users.
- Current system version.
- Number of daily visitors to the system.
- Number of trips made daily.
- Number of hazards identified in each trip.
- Number of people registering for the system daily.
- Number of clicks on advertisements.



The system must allow a user-friendly display like a **table or graph**, for example using an external graphing service.