Techne-Logos

Software Engineering Project

Pothole Detection System

Prepared by:

Luke Arjoon-816007674

Joshua Stapleton -816010544

Jose Bravo Mata -816011156

Neil Bhagwandeen- 816008609

Prepared for:

Ms. Shareeda Mohammed

**Table of Contents**

**Group Introduction……………………………………………………………………………3**

Group Members…………………………………………………………………………3

Group Meeting Times…………………………………………………………………...3

Collaboration Tools…………………………………………………………………......3

**Project Introduction…………………………………………………………………………...4**

Project Description……………………………………………………………………...4

Rational for Project…………………………………………………………………......4

User and System Requirements…………………………………………………………4

**Functional and Non-Functional Requirements………………………………………………8**

Functional Requirements & Non-functional Requirements …………………………….8

User Stories……………………………………………………………………………..11

**Use Cases……………………………………………………………………………………….15**

Simple Use Cases from User Stories……………………………………………………15

Use Case Diagram……………………………………………………………………….22

Ranking of Use Cases……………………………………………………………………23

Expanded Use Cases……………………………………………………………………..24

**Sequence Diagram……………………………………………………………………………...30**

**Class Diagram ………………………………………………………………………………….31**

**Risk Management………………………………………………………………………………32**

**Cost Estimation…………………………………………………………………………………34**

**Group Introduction**

**Group Members**

Luke Arjoon: Designer, Web programmer

Joshua Stapleton: Lead Documenter, Designer, Database Designer

Jose Bravo Mata: Designer, System Programmer

Neil Bhagwandeen: Designer, Project Lead, System Programmer

**Group Meeting Times**

Our team intends to hold internal meetings on Mondays and Wednesdays 10am - 12pm.

**Collaboration Tools**

In order to complete this project, our group intends to use certain tools which enable long-distance communication and collaboration. Some of these tools are outlined below.

*Skype*

Skype, a popular video chatting application, will be used when group members need to share a visual aspect of the project with his fellow members in real-time. For example, design or functionality changes to the project.

*WhatsApp*

At the most basic level, our group will be communicating and collaborating using popular social media platforms such as WhatsApp. This instant messaging platform help group members to share little tidbits of information in a timely fashion.

*Google Docs*

The Google Docs software will be used by our team for any documentation that needs to be done throughout the project. Google Docs make it simple for the multiple members of our team to contribute to a single text document simultaneously.

*GitHub*

We will use GitHub in order for each member of the team to be able to contribute to the project without necessarily having to download files to their local machine, or having to send files to their fellow members.

**Project Introduction**

**Project Description:**

This project aims to develop the roads of Trinidad and Tobago though the marrying of software and hardware components. The project is intended to be supplemented by the government of Trinidad and Tobago.

The system we propose will gather, and compute data obtained from accelerometers, cameras, a gyroscope and a series of sensors to create a pathway for accurate detection and maintenance of these potholes on the nation’s roadways. When the software detects a pot hole the coordinates will be sent to a server where server side of the application which will push it to a database where it will now be read from the database and place on a map of Trinidad and Tobago. There will be a primary and secondary method of pin pointing the pothole’s location. Firstly, by utilizing the Global Positioning System (GPS) to obtain these coordinates. The secondary method will be the use of the IP address of the vehicle’s current location. This final data will now be available to the stakeholders whereby they can readily deploy man power to fix the potholes with known locations. The potholes will be treated with priority where highways will be the highest followed by main roads and streets. This system will be designed to allow flexibility and improvement based on future requirements and tailored to benefit the citizens of Trinidad and Tobago.

**Rationale for the project:**

Throughout the nation of Trinidad and Tobago there are terrible roadways which are not being maintained due to no knowledge by the relevant authorities. A pothole is a type of failure in an asphalt pavement, caused by the presence of water in the underlying soil structure and the presence of traffic passing over the affected area. Our project will make a way for proper awareness of the state of the roadways and allow for good standard and consistency of the roads.

**User and System Requirements**

User Requirements Definition

1. The *Pothole Detection System* (PDS) must provide a means of representing the locations of the unfixed potholes for the general public.

System Requirements Specification

* 1. The system should index search through the database daily and pin the coordinates of potholes that have the status of “unfixed” to the map application.
  2. The system should check the status of the pinned locations daily to see if it has changed from unrepaired to fixed.
  3. If any locations had their status changed from “unfixed” to “fixed”, that location should be unpinned from the map.

User Requirements Definition

1. The PDS should be able to generate routes given start and end locations.

System Requirements Specification

* 1. Given a start location and end location, the system should generate multiple routes to the destination.
  2. The system should select the fastest route to the destination based on length and the route with the least number of potholes.
  3. The fastest route should be highlighted green and the route with the least number of potholes should be highlighted blue.

User Requirements Definition

1. The PDS must provide a means of using the database manager’s / dispatcher’s fingerprints to log into the database.

System Requirements Specification

* 1. The manager’s / dispatcher’s fingerprints should first be scanned using an ultrasonic fingerprint sensor where their fingerprint and their work ID would be stored as encrypted data in ROM storage.
  2. When the database manager / dispatcher wants to enter the database, he/she would enter their ID and scan their fingerprint.
  3. The system should first check to see if the ID matches any in the ROM storage and then compare both the stored fingerprint and the newly scanned fingerprint.
  4. If the fingerprint matches, the system should allow the user to enter the database else an appropriate message should be displayed along with a one-minute time where by the user would have to wait one minute to try to renter the database.

User Requirement Definition

1. The PDS must provide a process for the database manager / dispatcher to manually enter pothole locations and size into the database.

System Requirements Specification

* 1. When the database manager / dispatcher enters the database, a menu should be printed with one of the options being to enter a location into the database.
  2. The user should be able to enter a location and size of a pothole.
  3. The system should first search the database to see if the location is present with a status of “unfixed”.
  4. If the location is found with the status of “unfixed”, the system should not add the location into the database.
  5. Under any other conditions, the location should be added into the database along with a date and the status which should be set to “unfixed”.
  6. The system should output a message stating that the location was added.

User Requirement Definition

1. The PDS must provide a process for the database manager / dispatcher to set the status of a pothole from “unfixed” to “fixed”.

System Requirements Specification

* 1. When the database manager / dispatcher enters the database, a menu should be printed with one of the options being to change the status of a pothole from “unfixed” to “fixed”.
  2. The user should be able to enter a location of a pothole.
  3. The system should first search the database to see if the location is present with a status of “unfixed”.
  4. If the location is found with the status of “unfixed”, the system should change the status to “fixed”.
  5. Under any other condition, the system should output a message stating that the location is incorrect.

User Requirements Definition

1. The PDS must provide a process for the locations to be sorted by priority

System Requirements Specification

* 1. When a location is added into the server, either by automated entry by the surveying vehicle or manual entry by the database manager / dispatcher, the system should first search the database to see if the location is present with a status of “unfixed”.
  2. If the location is found with the status of “unfixed”, the system should not add the location into the database.
  3. Under any other conditions, the location should be added into the database along with a date and the status should be set to “unfixed”.
  4. The system should then check to see if the location is found to be on a highway, main roads and or streets.
  5. When the system determines the location, if the pothole is on the highway, that location will be set on a high priority alert followed by medium priority for a main road pothole, and then lowest priority for street pothole.
  6. The system should be able to use this priority strategy to sort the potholes.

User Requirements Definition

1. The PDS must be able to remove locations that are older than three years.

System Requirements Specification

* 1. At the last day of every month at 6:00pm, the system should go through the database searching for locations with a date older than three years from system date.
  2. If a location is found with a date older than three years, it then checks the status.
  3. If the status is “fixed”, the PDS should remove the location from the database.
  4. If the status is “unfixed”, the PDS should not remove the location from the database.
  5. This process should be repeated until the database is completely searched.

User Requirements Definition

1. The PDS must provide a daily backup of data on the surveying vehicle.

System Requirements Specification

* 1. When the array of sensors detects a pothole, the coordinates of the vehicle would be taken from a GPS module.
  2. This location is then stored along with a date on an onboard on a storage module.
  3. The coordinates are then encoded into a message and then sent to the server where the message is decoded, and location added into database.

User Requirements Definition

1. The PDS must allow for the retrieval of the location with the highest priority.

System Requirements Specification

* 1. When the database manager / dispatcher enters the database, a menu should be printed with one of the options being to retrieve the location of the pothole with the highest priority.
  2. As the database is sorted when a location is added, the computer retrieves the location at the top of the database and displayed to the database manager / dispatcher.

User Requirements Definition

1. The PDS must be able to differentiate between false and true readings.

System Requirements Specification

* 1. There are multiple sets of sensors under the surveying vehicle.
  2. If a sensor detects a pothole, the same pothole should be detected by at least two other sensors
  3. If two or more sensors also detect the pothole, then and only then it can be said that a pothole was found.

User Requirements Definition

1. The PDS must provide a process for the dispatcher to send the location of a pothole to a road repair team.

System Requirements Specification

* 1. When the database manager / dispatcher enters the database, a menu should be printed with one of the options being to send the location with the highest priority to the road repair team.
  2. As the database is continuously sorted, the computer retrieves the location and size at the top of the database, encodes it and sends it to the team leader of the road repair group.
  3. When the team leader of the road repair group retrieves the message, the location is then automatically resent to the server where error checking occurs. If the location is correct, the server should wait for an ack and a location else the server should resend the correct location to the road repair team.
  4. Once the server receives a ack and location, a message is displayed on database manager’s / dispatcher’s screen stating that the pothole at location has been fixed.

**Functional and Non-Functional Requirements**

**Functional and Non-Functional Requirements**

1)Functional Requirement:

The *Pothole Detection System* (PDS) must provide a means of representing the locations of the unfixed potholes for the general public.

2) Functional Requirement:

The PDS should be able to generate routes given start and end locations.

Non-Functional Requirements:

* The system must be able to generate the fastest route(distance), and the route the route with the least number of potholes within 10 seconds to generate a map from the time the user enters the routes.
* The app must be able to get the route within five inputs of the user

3)Functional Requirement:

The PDS should allow for the general public to call and report a pothole that is not

seen in the application.

Non-Functional Requirements

* The caller should relate to a dispatcher / database manager within 1 minute of placing the call.

4)Functional Requirement:

The PDS must provide a means of using the database manager’s / dispatcher’s fingerprints to log into the database.

Non-Functional Requirements

* It should take the user no more than 5 seconds to log into the database.
* After 15 minutes of inactivity the database should log the user out of the system.
* After a failed attempt, the user should have wait one minute until being able to enter login credentials again.
* After 5 failed attempts, the user should have wait 30 minutes until being able to enter login credentials again.

5)Functional Requirement:

The PDS must provide a process for the database manager / dispatcher to manually enter pothole locations and size into the database.

Non-Functional Requirements

* The dispatcher should be able to enter information on potholes within 4 steps.
* The system should check if the location is valid i.e. the location is in Trinidad and Tobago.
* It should take one minute in total for the location to be added into the database and for the database to be resorted.

6)Functional Requirement:

The PDS must provide a process for the database manager / dispatcher to set the status of a pothole from “unfixed” to “fixed”.

Non-Functional Requirements

* The dispatcher should be able to edit the status of the potholes in less than 4 steps.
* The database should be update itself in less than 15 seconds.

7)Functional Requirement:

The PDS must provide a process for the locations to be sorted by priority.

Non-Functional Requirements

* It should take the database no more than 30 seconds the locations to be sorted by priority.

8)Functional Requirement:

The PDS must be able to remove locations that are older than three years.

Non-Functional Requirements

* The system should be able to automatically remove the fixed potholes which are older than three years and have a status of fixed on the last day of every month within 15 minutes no later than 8:00pm

9)Functional Requirement:

The PDS must provide a daily backup of data on the surveying vehicle.

Non-Functional Requirements

* The system should store the location and size of a pothole 5 seconds after it is found.
* At 8:30am the next day, the locations and size of the potholes found the day before can be overwritten by the new day’s pothole information.

10)Functional Requirement:

The PDS must allow for the retrieval of the location with the highest priority.

Non-Functional Requirements

* The dispatcher must be able to retrieve the location with the highest priority in 3 steps.
* The database should be able to retrieve this in less than 30 seconds.

11)Functional Requirement:

The PDS must be able to differentiate between false and true readings.

Non-Functional Requirements

* The sensors on the vehicle should simultaneously be able to gather information on the pothole in one to five seconds (depending on size of the pothole).
* The sensors should also be able to compare the data retrieved with each other in five seconds.
* The system should be able to send the pothole information to the server in 20 seconds.

12)Functional Requirement:

The PDS must provide a process for the dispatcher to send the location of a pothole to a road repair team.

Non-Functional Requirements

* The dispatcher will be able to send a location to the repair team in no more than 4 steps.
* The repair team should get the alert in less than 5 minutes.
* The system should take no more than 1 minute to verify the location sent.
* The repair team leader should be able to send the acknowledgement for completion of job in no more than 4 steps and it should take no longer than 2 minutes for the message to reach the server.
* The database should be able to process acknowledgement and inform database manager to change status of a pothole in 1 minute from receiving acknowledgement.

**User Stories**

**Scenarios**

Scenario name : drivingToWork

Actor instances: Jeff: End user

Flow of events :

1. Jeff is driving to work, Josh is alerted by the pothole application and notices that there is a newly formed pothole approximately one km away.
2. He adjusts his speed and can slowly and safely cross the pothole.

Scenario name : goingToTheBeach

Actor instances: Tim: End user

Flow of events :

1. Tim wants to go to the beach, he opens the application and enters his destination.
2. The application, taking into consideration the number of potholes, displays a route for Tim that is the fastest and with the least number of potholes

Scenario name : unreportedPothole

Actor instances: Jeffery: End user, Bob: Database manager

Flow of events :

1. Jeffery is walking to the mini mart and he notices that there is a pothole in the road.
2. Jeffery opens the application and searches to see if the pothole is pinned on the map and notices that it was not.
3. Jeffery calls the hotline and makes a report with Bob.
4. Bob requests that Jeffery give him his location and the approximate dimension of the pothole.
5. Bob receives the location and approximate size and enters it into the database which sorts the location according to priority.

Scenario name : surveying

Actor instances: Dale: Surveyor

Flow of events :

1. Dale is driving in the surveying vehicle and sees a pothole ahead, Dale drives over the pothole and the equipment processes the size of the pothole
2. The equipment connects to the server and sends its current location the size of the pothole and stores a copy of the data onboard.
3. The server receives the location and size, adds a date and sorts the location according to priority.

Scenario name : noNewPothole

Actor instances: Dale: Surveyor, Bob: database manager / dispatcher

Flow of events :

1. Bob uses his finger print to log into the server, Bob sees that no pothole location was sent to the server since yesterday, Bob calls Dale and requests ack. (request if Dale encountered any potholes)
2. Dale receives request and sends nack.
3. Bob receives nack and logs out of server.

Scenario name : missingFromTheServer

Actor instances: Dale: Surveyor, Bob: database manager / dispatcher

Flow of events :

1. Bob uses his finger print to log into the server, Bob sees that no pothole location was sent to the server since yesterday, Bob calls Dale and requests ack. (request if Dale encountered any potholes)
2. Dale receives request and send ack.
3. Bob requests that Dale return to headquarters before the end of the day.
4. Dale receives request and returns to headquarters.
5. Bob retrieves the data from the equipment and manually enters the location into the server where a date is attached, and the location is sorted by priority.

Scenario name : troubleWithSurveyingVehicle

Actor instances: Jack: Surveyor

Flow of events :

1. Jack is about to leave to begin surveying for potholes, he tries to start his vehicle but to no avail.
2. He quickly removes the equipment from one vehicle and attaches it to his other vehicle and sets out to scan the roads for potholes.

Scenario name : accident

Actor instances: Kim, Tim: Surveyor

Flow of events :

1. Kim was surveying the road when a car crashed into her, she called Tim another surveyor.
2. Tim arrived and removed the equipment from Kim’s vehicle and carried it back to the headquarters for it to be installed on another vehicle.

Scenario name : dispatching

Actor instances: John: Database manager / dispatcher,

Josh: road repair team leader

Flow of events :

1. John uses his fingerprint to enter into the database, he sends Josh the location and size of the pothole with the highest priority and awaits a response of a location.
2. Josh receives the location and size, sends the location back to John and heads to the location to repair the pothole.
3. John receives location, ensures that the location is correct and waits for an ack and location.
4. After completing the road repair, Josh sends ack and location to John.
5. John receives ack and location and sets the status of pothole location as fixed.

Scenario name : wrongLocation

Actor instances: John: Database manager / dispatcher,

Josh road repairman team leader

Flow of events :

1. John uses his fingerprint to enter into the database, he sends Josh the location and size of the pothole with the highest priority and awaits a response of a location.
2. Josh receives an incorrect location and size, sends the incorrect location back to John and heads to the location to repair the pothole.
3. John receives location, sees that the location is incorrect and resends the location and size of the pothole.
4. Josh receives the location and size, sends the location back to John and heads to the new location to repair the pothole.
5. John receives location, ensures that the location is correct and waits for an ack and location.
6. After completing the road repair, Josh sends ack and location to John.
7. John receives ack and location and sets the status of pothole location as fixed.

Scenario name : reoccuringPothole

Actor instances: Kelly: Database manager / dispatcher,

Dane: road repairman team leader

Flow of events :

1. Kelly uses her fingerprint to enter into the database, she notices that the location with the highest priority already had work done to it.
2. Kelly sends Dane the location and size of the pothole she also informs him that previous work had been done to it and awaits a response of a location.
3. Dane receives the location, size and details, sends the location back to Kelly and heads to the location to repair the pothole
4. Kelly receives location, ensures that the location is correct and waits for an ack and location.
5. Dane arrives at the pothole, performs a detailed examination and fixes the pothole.
6. Dane sends an ack and the location to Kelly.
7. Kelly receives ack and location and sets the status of the pothole location as fixed.

**Use Cases**

**Simple Use Cases**

|  |  |  |
| --- | --- | --- |
| USE CASE NAME | Views potholes | |
| ACTORS | End user | |
| INPUTS | - | |
| OUTPUTS | Map with unfixed potholes pinned to it | |
| FLOW OF EVENTS | ACTOR | SYSTEM |
| 1. The user opens the application | 1. The application does an index search through the database and look for potholes with the status unfixed 2. It then pins the location of the potholes to the map |
| EXEPTIONS | - | |
| COMMENTS | - | |

Table 1: Showing use case for viewing potholes.

|  |  |  |
| --- | --- | --- |
| USE CASE NAME | Views fastest route and route with least potholes | |
| ACTORS | End user | |
| INPUTS | Start and end locations | |
| OUTPUTS | Fastest route and route with least potholes | |
| FLOW OF EVENTS | ACTOR | SYSTEM |
| 1. The user enters start and end location | 1. The application generates multiple routes to the destination 2. The application selects the fastest route to the destination and highlight it in green 3. The application counts the number of potholes with the status “unfixed” on each route and display the route with the least number of potholes highlighted in blue |
| EXEPTIONS | Alt-Step 4: The fastest route is also the route with least number of potholes | |
| COMMENTS | - | |

Table 2: Showing use case for viewing fastest and route with least potholes.

|  |  |  |
| --- | --- | --- |
| USE CASE NAME | Log in | |
| ACTORS | Dispatcher / Database Manager(DM) | |
| INPUTS | Work ID and Fingerprint | |
| OUTPUTS | Menu interface | |
| FLOW OF EVENTS | ACTOR | SYSTEM |
| 1. The user enters hi work ID and scans his fingerprint | 1. The PDS checks to see if the ID matches any in the ROM storage and then compare both the stored fingerprint and the newly scanned fingerprint 2. Once work ID and fingerprints are correct, the PDS allows the user to enter. |
| EXEPTIONS | Alt-Step 3: Work ID or fingerprint is incorrect | |
| COMMENTS | - | |

Table 3: Showing use case for logging in.

|  |  |  |
| --- | --- | --- |
| USE CASE NAME | Add location | |
| ACTORS | DM | |
| INPUTS | Location and size | |
| OUTPUTS | Location added | |
| FLOW OF EVENTS | ACTOR | SYSTEM |
| 1. The user enters location and size | 1. The PDS searches to see if the location is recorded with a status of “unfixed”. 2. Once found, the PDS outputs location added |
| EXEPTIONS | Alt-Step 3: Under any other condition, the location is given a priority and is added to the PDS where an output of location added is displayed | |
| COMMENTS | - | |

Table 4: Showing use case for adding location.

|  |  |  |
| --- | --- | --- |
| USE CASE NAME | Change pothole status | |
| ACTORS | DM | |
| INPUTS | Location | |
| OUTPUTS | Status changed | |
| FLOW OF EVENTS | ACTOR | SYSTEM |
| 1. The user enters location | 1. The PDS searches to see if the location is recorded with a status of “unfixed”. 2. Once found, the PDS changes the status of the location from “unfixed” to “fixed” |
| EXEPTIONS | Alt-Step 3: Under any other condition, the PDS outputs that the location is incorrect | |
| COMMENTS |  | |

Table 5: Showing use case for changing pothole status.

|  |  |  |
| --- | --- | --- |
| USE CASE NAME | Dispatch repair team | |
| ACTORS | DM | |
| INPUTS | - | |
| OUTPUTS | Location sent to road repair team | |
| FLOW OF EVENTS | ACTOR | SYSTEM |
| 1. The Dispatcher / Database Manager selects the option to dispatch a road repair team | 1. The PDS responds by selecting the pothole location with the highest priority 2. The PDS then encodes the location and sends it to the leader of the repair team 3. The PDS then waits for the automatic response from the repair team where an error check of the data occurs 4. The PDS sends ack to repair team once there are no errors |
| EXEPTIONS | Alt-Step 3: The location which was sent by the PDS to the repair team became corrupted during transmission which caused the automatic response from the team to be incorrect.  Alt-Step 4: The expected automatic response from the repair team became corrupted during transmission  Alt-Step 5: Once the error check is completed and the PDS determines that the response location is not the same as the one transmitted, the encoded location is resent to the repair team and the system awaits for another automatic response. | |
| COMMENTS |  | |

Table 6: Showing use case for dispatching road repair team.

|  |  |  |
| --- | --- | --- |
| USE CASE NAME | Send location to PDS | |
| ACTORS | Sensor array | |
| INPUTS | Pothole detected | |
| OUTPUTS | - | |
| FLOW OF EVENTS | ACTOR | SYSTEM |
| 1. The sensor array detects a pothole. | 1. The detecting system responds by taking the coordinates from the GPS module 2. The detecting system then calculates average size. 3. The detecting system encodes the location and size of the pothole and sends it to the PDS. 4. The detecting system then waits for the automatic response from the PDS where an error check of the data occurs 5. The detecting system sends ack PDS once there are no errors 6. ( use case Store location on-board) |
| EXEPTIONS | Alt-Step 4: The location which was sent by the detecting system to the PDS became corrupted during transmission which caused the automatic response from the PDS to be incorrect.  Alt-Step 5: The expected automatic response from the PDS became corrupted during transmission  Alt-Step 7: Once the error check is completed and the detection system determines that the response location is not the same as the one transmitted, the encoded location is resent to the PDS and the system awaits for another automatic response. | |
| COMMENTS |  | |

Table 7: Showing use case for sending location to PDS.

|  |  |  |
| --- | --- | --- |
| USE CASE NAME | Store location on-board | |
| ACTORS | Sensor Array | |
| INPUTS | Pothole detected | |
| OUTPUTS | - | |
| FLOW OF EVENTS | ACTOR | SYSTEM |
| 1. The sensor array detects a pothole. | 1. (use case send location to PDS) 2. The detecting system checks if location has already been recorded for the day. 3. Once not recorded, the location and size is stored in the systems daily log. |
| EXEPTIONS | Alt-Step 4: the location is already in daily log. | |
| COMMENTS |  | |

Table 8: Showing use case for storing location onboard.

|  |  |  |
| --- | --- | --- |
| USE CASE NAME | Send repaired location | |
| ACTORS | Road repair team lead | |
| INPUTS | - | |
| OUTPUTS | Location sent for status change | |
| FLOW OF EVENTS | ACTOR | SYSTEM |
| 1. The repair team lead selects the option to send fixed location to Dispatcher / Database Manager | 1. The application responds by encoding the pothole location sent by the Dispatcher / Database Manager 2. The application then sends the location to the PDS 3. The application then waits for the automatic response from the PDS where an error check of the data occurs 4. The application sends ack PDS once there are no errors 5. The location is removed from the application |
| EXEPTIONS | Alt-Step 3: The location which was sent by the application to the PDS became corrupted during transmission which caused the automatic response from the PDS to be incorrect.  Alt-Step 4: The expected automatic response from the PDS became corrupted during transmission  Alt-Step 5: Once the error check is completed and the application determines that the response location is not the same as the one transmitted, the encoded location is resent to the PDS and the system awaits for another automatic response. | |
| COMMENTS |  | |

Table 9: Showing use case for sending repaired location to PDS.

**Use Case Diagram**

**A close up of a map

Description automatically generated**

Figure 1: Showing use case diagram derived from simple use cases.

**Ranking of Use Cases**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Use-Case Name | Ranking Criteria 1-5 | | | | | | Total | Priority | Build Cycle |
|  | 1 | 2 | 3 | 4 | 5 | 6 |  | | |
| View Pothole | 2 | 2 | 2 | 1 | 2 | 1 | 10 | Low | 5 |
| Views fastest route and route with least potholes | 2 | 2 | 2 | 1 | 2 | 1 | 10 | Low | 5 |
| login | 3 | 3 | 2 | 4 | 2 | 2 | 16 | Medium | 4 |
| Add location | 4 | 3 | 3 | 2 | 2 | 2 | 16 | Medium | 3 |
| Change pothole status | 4 | 3 | 3 | 2 | 3 | 3 | 18 | Medium | 3 |
| Dispatch repair team | 5 | 5 | 5 | 3 | 4 | 5 | 27 | High | 1 |
| Send location to server | 5 | 5 | 4 | 3 | 4 | 5 | 26 | High | 2 |
| Store location on-board | 3 | 3 | 3 | 2 | 2 | 2 | 15 | Medium | 4 |
| Send repaired location | 5 | 5 | 5 | 3 | 4 | 5 | 27 | High | 1 |

Table 10: Showing ranking of use cases.

1. Significant impact on the architectural design.

2. Easy to implement but contains significant functionality.

3. Includes risky, time-critical, or complex functions.

4. Involves significant research or new or risky technology.

5. Includes primary business functions.

6. Will increase revenue or decrease costs.

1-10 inclusive low

11-20 inclusive medium

21-30 inclusive high

**Expanded Use Cases**

|  |  |  |
| --- | --- | --- |
| USE CASE NAME | Dispatch repair team | |
| USE CASE ID | UC-10001 | |
| PRIORITY | High | |
| SOURCE |  | |
| PIRMARY BUSINESS ACTOR | Dispatcher / Database Manager | |
| OTHER PARTICAPATING ACTORS | Road repair team lead | |
| DESCRIPTION | This use case describes the event of a Dispatcher / Database Manager dispatching a road repair team to a pothole location. The Dispatcher / Database Manager selects the option to dispatch a road repair team where the PDS sends the location with the highest priority to the repair team | |
| PRECONDITION | The Dispatcher / Database Manager is already logged into the PDS | |
| TRIGGER | This use case is initiated when the Dispatcher / Database Manager selects the option to dispatch a road repair team | |
| TYPICAL COURSE OF EVENTS | ACTOR ACTION | SYSTEM RESPONSE |
| 1. The Dispatcher / Database Manager selects the option to dispatch a road repair team | 1. The PDS responds by selecting the pothole location with the highest priority 2. The PDS then encodes the location and sends it to the leader of the repair team 3. The PDS then waits for the automatic response from the repair team where an error check of the data occurs 4. The PDS sends ack to repair team once there are no errors |
| ALTERNATE COURSES | Alt-Step 3: The location which was sent by the PDS to the repair team became corrupted during transmission which caused the automatic response from the team to be incorrect.  Alt-Step 4: The expected automatic response from the repair team became corrupted during transmission  Alt-Step 5: Once the error check is completed and the PDS determines that the response location is not the same as the one transmitted, the encoded location is resent to the repair team and the system awaits for another automatic response. | |
| CONCLUSION | This use case concludes when the system sends a ack to the team. | |
| POST CONDITION | The road repair team receives the pothole location | |
| BUSINESS RULES | The repair team can only receive a new location once they have reported the pervious pothole fixed. | |
| IMPLIMENTATION CONSTRAINTS AND SPECIFICATION | Menu to be provided to the Dispatcher / Database Manager and an application to be provided for the repair team. | |
| ASSUMPTIONS | The Dispatcher / Database Manager would be notified on a successful transmission of location. | |

Table 11: Showing expended use case for dispatch road repair team.

|  |  |  |
| --- | --- | --- |
| USE CASE NAME | Send repaired location | |
| USE CASE ID | UC-10002 | |
| PRIORITY | High | |
| SOURCE |  | |
| PIRMARY BUSINESS ACTOR | Road repair team leader | |
| OTHER PARTICAPATING ACTORS |  | |
| DESCRIPTION | This use case describes the event of the Road repair team leader sending a fixed pothole location to the Dispatcher / Database Manager for the status of the pothole to be changed to fixed. The repair team lead selects the option to send location off fixed pothole where the application sends the location to the Dispatcher / Database Manager. | |
| PRECONDITION | The repair team lead received a location and has repaired the pothole | |
| TRIGGER | This use case is initiated when the repair team lead selects the option to send fixed location to Dispatcher / Database Manager | |
| TYPICAL COURSE OF EVENTS | ACTOR ACTION | SYSTEM RESPONSE |
| 1. The repair team lead selects the option to send fixed location to Dispatcher / Database Manager | 1. The application responds by encoding the pothole location sent by the Dispatcher / Database Manager 2. The application then sends the location to the PDS 3. The application then waits for the automatic response from the PDS where an error check of the data occurs 4. The application sends ack PDS once there are no errors 5. The location is removed from the location |
| ALTERNATE COURSES | Alt-Step 3: The location which was sent by the application to the PDS became corrupted during transmission which caused the automatic response from the PDS to be incorrect.  Alt-Step 4: The expected automatic response from the PDS became corrupted during transmission  Alt-Step 5: Once the error check is completed and the application determines that the response location is not the same as the one transmitted, the encoded location is resent to the PDS and the system awaits for another automatic response. | |
| CONCLUSION | This use case concludes when the location is removed from the application. | |
| POST CONDITION | The Dispatcher / Database Manager receives the location | |
| BUSINESS RULES | The repair team can only receive a new location once they have reported the pervious pothole fixed. | |
| IMPLIMENTATION CONSTRAINTS AND SPECIFICATION | Menu to be provided to the Dispatcher / Database Manager and an application to be provided for the repair team. | |
| ASSUMPTIONS | The repair team lead would be notified on a successful transmission of location. | |

Table 12: Showing expended use case for send repaired location.

|  |  |  |
| --- | --- | --- |
| USE CASE NAME | Send location to PDS | |
| USE CASE ID | UC-10003 | |
| PRIORITY | High | |
| SOURCE |  | |
| PIRMARY BUSINESS ACTOR | Sensor array | |
| OTHER PARTICAPATING ACTORS |  | |
| DESCRIPTION | This use case describes the event of the Sensor array sending a pothole location to the PDS for the location of the pothole to be given a priority and stored. When the array of sensors automatically detects a pothole, the current location and size is sent to the PDS. | |
| PRECONDITION | The sensor array is on and has completed its automatic calibration | |
| TRIGGER | This use case is initiated when the sensor array detects a pothole | |
| TYPICAL COURSE OF EVENTS | ACTOR ACTION | SYSTEM RESPONSE |
| 1. The sensor array detects a pothole. | 1. The detecting system responds by taking the coordinates from the GPS module 2. The detecting system then calculates average size. 3. The detecting system encodes the location and size of the pothole and sends it to the PDS. 4. The detecting system then waits for the automatic response from the PDS where an error check of the data occurs 5. The detecting system sends ack PDS once there are no errors 6. (use case Store location on-board) |
| ALTERNATE COURSES | Alt-Step 4: The location which was sent by the detecting system to the PDS became corrupted during transmission which caused the automatic response from the PDS to be incorrect.  Alt-Step 5: The expected automatic response from the PDS became corrupted during transmission  Alt-Step 7: Once the error check is completed and the detection system determines that the response location is not the same as the one transmitted, the encoded location is resent to the PDS and the system awaits for another automatic response. | |
| CONCLUSION | This use case concludes when the location and size is stored on board. | |
| POST CONDITION | The PDS assigns a priority to the location and is then stored | |
| BUSINESS RULES |  | |
| IMPLIMENTATION CONSTRAINTS AND SPECIFICATION |  | |
| ASSUMPTIONS | The sensor array would have a constant consistent network connection | |

Table 13: Showing expended use case for send location to PDS.

**Sequence Diagram**

**Top use case**

**A close up of a map

Description automatically generated**

Figure 2: Showing sequence diagram for top use case.

**Class Diagram**

**A close up of text on a white background

Description automatically generated**

Figure 3: Showing class diagram.

**Risk Management**

**Risk Management**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Risk** | **Probability** | **Effect** | **Affect** | **Strategy Category** | **Strategy** |
| Staff  turnover | Low | Tolerable | Project | Avoidance | Implement pair programming |
| Requirements change | Moderate | Serious | Project and product | Minimization | Implement extreme programming techniques. Project requirements will be left open for alterations. |
| Lack of conflict resolution | Low | Catastrophic | Project | Avoidance | Maintain high level of communication within the team. |
| Wrong Estimation and Scheduling | Moderate | Serious | Project and product | Avoidance | Keep members informed about the details of each stage of development. |
| Breakdown of specification | Low | Catastrophic | Project and product | Avoidance | Ensure the fundamental aspects of the project are outlined from the start. |
| Productivity issues | Moderate | Catastrophic | Project | Avoidance | Constantly inform members of their roles and responsibilities. |
| No proper subject training | Moderate | Serious | Project and product | Avoidance | Use small, well trained team that is familiar with the requirements and programming languages involved. |
| Insufficient resources | High | Catastrophic | Project, product and business | Minimization | Acquire approval and funding from the ministry. Maintain a proper financial structure of the budget. |
| Product too complex | Moderate | Catastrophic | Project, product and business | Contingency | Review specifications and divide tasks into even smaller parts. |

Table 14: Showing Risk, Probability, Effect, Affect, Strategy Category and Strategy.

**Cost estimation**

**Modelling Technique Used**

The main cost modelling technique used in the software pricing component of this project would be COCOMO II (Constructive Cost Model).

**Rationale for using COCOMO II**