Department of Computer Science

Shaheed Rajguru College of Applied Sciences for Women

Software Engineering Project Report

**DTC Bus Tracking System**

Submitted by:

Shrutika Shaw (2019345)

Shuchi Sharma (2019346)

Vanitha K (2019358)

**Certificate**

This is to certify that the project entitled ‘DTC Bus Tracking System’ is the work carried out by Shrutika Shaw, Shuchi Sharma, Vanitha K of B.Sc (Hons) Computer Science at Shaheed Rajguru College of Applied Sciences for Women. This report has not been submitted to any other organisation/ institute for the award of any other Degree/ Diploma.

Teacher Mentor Principal

Ms. Urmil Bharti Dr. Payal Mago

Students

Shrutika Shaw

Shuchi Sharma

Vanitha K

**Acknowledgement**

We have taken efforts in this project. However, it would not have been possible without the support and help of many individuals. We would like to extend our sincere thanks to all of them.

We are highly indebted to Ms. Urmil Bharti for her guidance and constant supervision as well as for providing necessary information regarding the project and also for her support for completing the project.

Our thanks and an appreciation also goes to our friends in developing the project and people who have willingly helped us out with their abilities.

**Index**

1. Problem Statement
2. Process Model
3. Hardware and Software Specification
   1. Developer End
   2. User End
4. Data Flow Diagram and Data Dictionary
   1. Level 0 DFD
   2. Level 1 DFD
   3. Data Dictionary
5. Architectural Design
6. Use Cases
   1. Use Case Diagram
7. Sequence Diagram
8. Estimation
   1. Function Point
      1. Value Adjustment Factors
      2. Computation of Function Points
   2. Effort Estimation
9. Testing
   1. Pseudo Code for Searching Bus
   2. Flow Graph
   3. Calculating Cyclomatic Complexity
   4. Independent Path
10. Risk Analysis
    1. Risk Table
11. Timeline Chart
12. Interfaces
    1. Passenger End
    2. Driver End
13. Bibliography

1.**Problem Statement**

Management of buses of public transportation systems is the main problem nowadays. Based on the current system, there is no such system which provides information about the DTC buses, its expected arrival time and the expected waiting time. Delhi transportation corporation (DTC) is the main public transport operator of Delhi. DTC operates on many routes in Delhi and neighbouring states which aims to provide efficient, economical, reliable and properly coordinated road transport systems. In the context of a properly coordinated transport system, the majority of the passengers prefer to use buses, but there are some issues with the arrival and departure of buses from their respective areas, because of that they have to waste their valuable time standing on the bus stops and waiting for the bus. Sometimes, buses reach to the empty bus stands, which wastes a lot of time and energy. So, neither do people know when their particular bus will arrive at their bus stands. Although the DTC bus system has a specific time table usually this service doesn’t follow this which leads to unreliability on DTC buses and passengers don’t prefer to travel through buses and therefore prefer their own vehicle which is leading to uncontrollable pollution in the city.

Lack of a real time platform is the serious communication problem between the bus users and the bus management team. Without a real time platform, the bus management side is unable to update the latest bus traffic information for passengers. Passengers also cannot check on the updated bus schedule if there is a bus delay.

Our application provides the relevant information regarding all the bus numbers going from user’s source & destination along with the route details and real time location. Our application is a two way application: one for the passengers where passengers will know the location of the buses in their vicinity and one for the bus system where the bus system can update its location.

Our app will have the following features:

For Passengers:

1. Track Location : Any passenger availing our services would be able to locate the buses from a particular source to a particular destination.

2. Searching: This will provide the Users with the list of Buses running from that particular Source to a Particular Destination. It will also allow the user to sort the data as per the time slots.

3. Notifications: The app will require permission from the User regarding the notification. Users will be updated and served with frequent notifications via Push Notifications Services, SMS, and Email. Notifications that will be sent would have information regarding:

• Bus Arrival

• Bus Delay

For Bus:

• Update Live Location of the Bus.

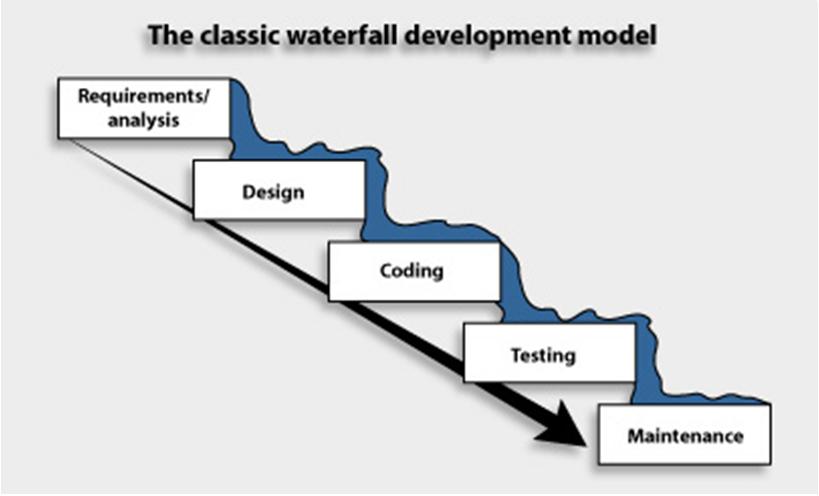
• Time for the Bus to reach the Particular Bus Stand.

2.**Process Model**

A software process model depicts the order in which the activities of the software development will be undertaken. It describes the sequence in which phases of the software lifecycle will be performed.

We have used Waterfall Model.

What is Waterfall Model?



The waterfall model is a classical model used in system development life cycle to create a system with a linear and sequential approach. It is termed as waterfall because the model develops systematically from one phase to another in a downward fashion. This model is divided into different phases and the output of one phase is used as the input of the next phase. Every phase has to be completed before the next phase starts and there is no overlapping of the phases

Waterfall Model in DTC BUS TRACKING SYSTEM

In DTC Bus Tracking System, it is easy to develop an overall plan for managing a Waterfall project because every phase has a specific start and end. This allows the team to ascertain exactly what coding is to be developed, when it’s due and when testing should begin. This can all be done before the commencement of the project. We use this model because the requirements are fixed and work is to proceed to completion in a linear manner

**3.Software requirement specification**

* Developer End:

●Operating System Requirements: Ubuntu(Linux), MacOS, Windows; Recommended Windows 10

●CPU: Intel or AMD processor with 64-bit support; Recommended: 2.8 GHz or faster processor

●Disk Storage: 4 GB of free disk space.

●Software requirements:

1. Node.js (version 14.1.0)

2. React-native CLI

3. Python (version- 2.7.18)

4. Android SDK and platform-tools

5. Android Studio for testing the application (Optional)

P.S: You can use an Android device with “developer” mode enabled.

6. Visual Studio Code (editor)

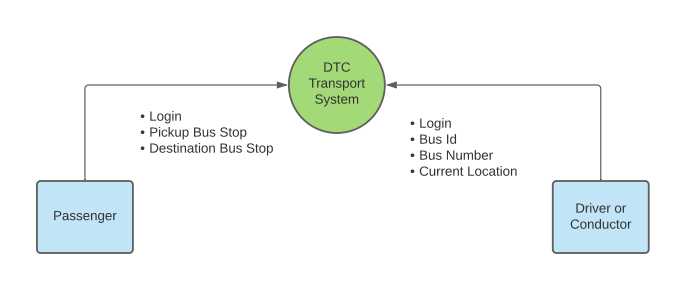
* User End:

1. Android Device

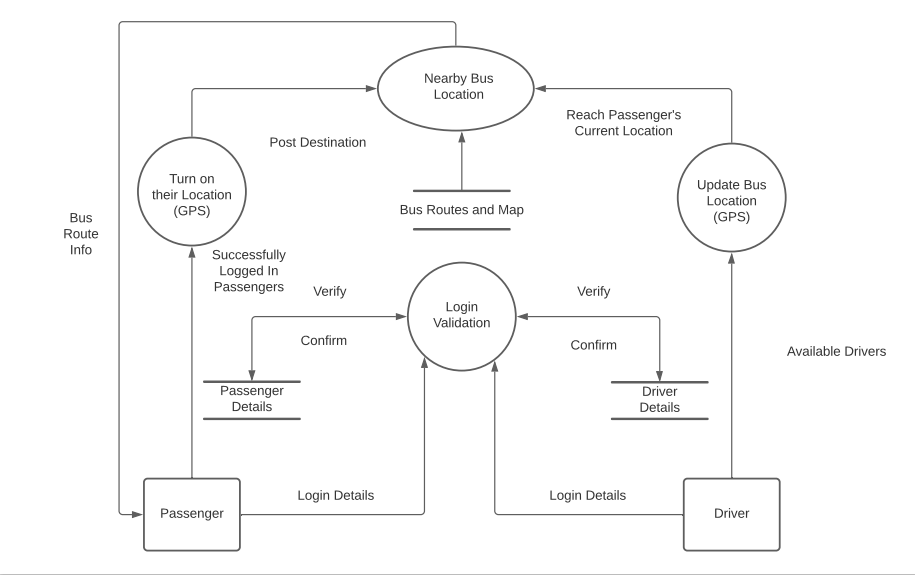
2. Internet

**4.Data Flow Diagram and Data Dictionary**

**4.1. Level 0 DFD**



**4.2. Level 1 DFD**



**4.3. Data Dictionary**

1. Sign up ->Name + Email ID + Phone number + Password

Confirm Password

2. Login ->Email ID + Password

3. Route Information ->Start Station + End Station + Route +

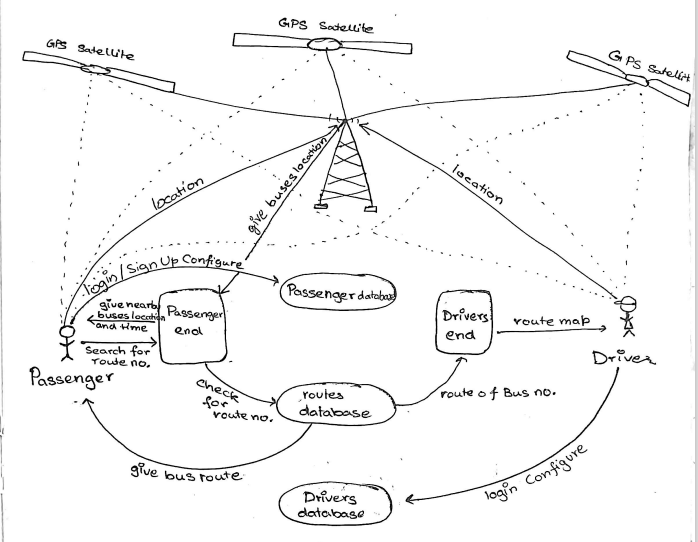
[Distance between two station | Cost | Time Taken | No of Station]

4. Bus Route -> [Route list]

5. DTC Bus MAP ->Image map of DTC Bus System

6. Bus Nearby -> [Location | Route number]+ GPS system

**5. Architectural Design**



**6. Use Cases**

***Use Case 1***: Search a Bus

*Primary Actor:* Passenger

*Precondition:* Passenger has Logged In

*Main Success Scenario:*

1. Passenger posts Location.
2. Passenger posts starting and ending Bus Stands.
3. Passenger finds the available Buses.

*Exception Scenarios:*

1. No buses available.

* Passenger gets notified.

***Use Case 2:*** Drop Passenger at Location

*Primary Actor:* Driver

*Precondition:* Driver has Logged In

*Main Success Scenario:*

1. Driver posts Location.
2. Driver gets Passenger’s Location.
3. Driver drops Passenger at Location.

*Exception Scenario:*

1. No request for Pickup.

* Driver gets notified.

**6.1. Use Case Diagram**

PASSENGER

TRACK LOCATION

SEARCHING

LOGIN

h









USE CASE DIAGRAM For DTC BUS SYSTEM

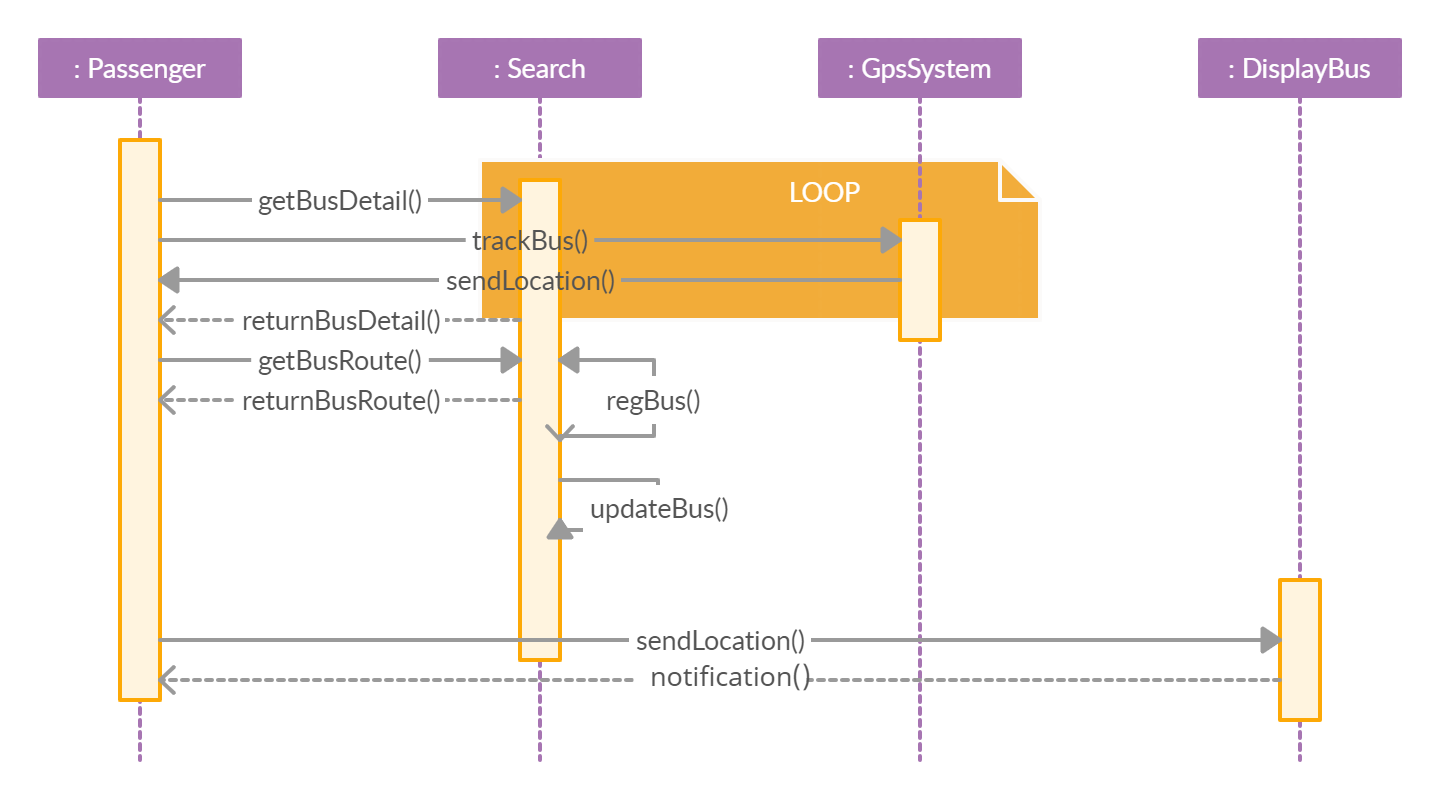
NOTIFICATION

UPDATE LOCATION

DRIVER



**7. Sequence Diagram**



**8. Estimation**

Function Point

The Function Point is a unit of measurement which expresses the amount of business functionality delivered by the information system. Function Point measures the size of the software.

To compute function points (FP), the following relationship is used:

**FP = count total \* [0.65 +0.01 \* ∑( Fi )]**

**8.1.1 Value Adjustment Factor**

|  |  |  |
| --- | --- | --- |
| S.no | Questions | Marks |
| 1. | Does the system require reliable backup and recovery? | 3 |
| 2. | Are specialized data communications required to transfer information to and from the application? | 3 |
| 3. | Are their distributed processing functions? | 1 |
| 4. | Is this performance critical? | 1 |
| 5. | Will the system run in an existing, heavily utilised operational environment? | 4 |
| 6. | Does the system require only one data entry? | 4 |
| 7. | Does the online data entry require the input transaction to be built over the multiple screens or operations? | 3 |
| 8. | Are the ILFs updated online? | 1 |
| 9. | Are the inputs, outputs, files or inquiries complex? | 2 |
| 10. | Is the internal processing complex? | 2 |
| 11. | Is the code designed to be reusable? | 3 |
| 12. | Are the conversion and installation included in the design? | 3 |
| 13. | Is the code designed for multiple installations in different organisations? | 2 |
| 14. | Is the application designed to facilitate change and ease of use by the user? | 4 |
|  | TOTAL | **36** |

∑(Fi) = 3 + 3 + 1 + 1 + 4 + 4 + 3 + 1 + 2 + 2 + 3 + 3 + 2 + 4

= 36

**8.1.2** **Computation of Function Points**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **INFORMATION DOMAIN VALUE** | **COUNT** |  | **SIMPLE** | **AVERAGE** | **COMPLEX** |  |
| User Input | 2 | 3 | 3 | **4** | 6 | 8 |
| User Output | 6 | 3 | 4 | **5** | 7 | 30 |
| User Inquiries | 3 | 3 | 3 | **4** | 6 | 12 |
| Internal Logical Files | 1 | 3 | 7 | **10** | 15 | 10 |
| External Interface | 2 | 3 | 5 | **7** | 10 | 14 |
| **COUNT TOTAL** |  | | | | | **74** |

As complexity adjustment factor is average for DTC bus tracking System.

Count total = 74

FP = 74 \* [0.65 + 0.01 \* 36]

= 74\* 1.01

= 74.74 ≈ 75

* 1. **Effort and Cost Estimation**

EFFORT = FP/PRODUCTIVITY

FP = 75

PRODUCTIVITY = 6

Effort =75/6

= 12.5 person.month ≈ 13 PM

COST = Effort \* Rate

Rate = $ 3000 PM

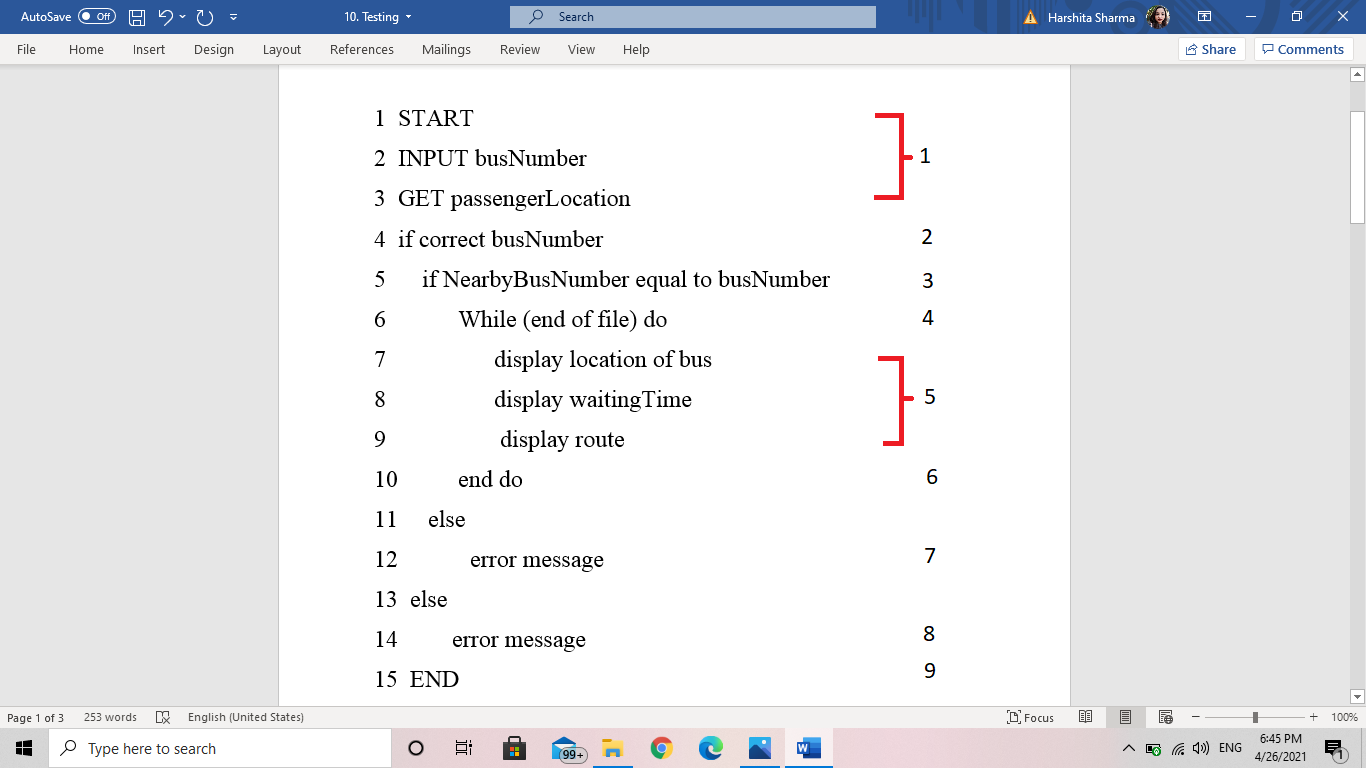
Effort = 13 PM

Cost = 13 \* 3000

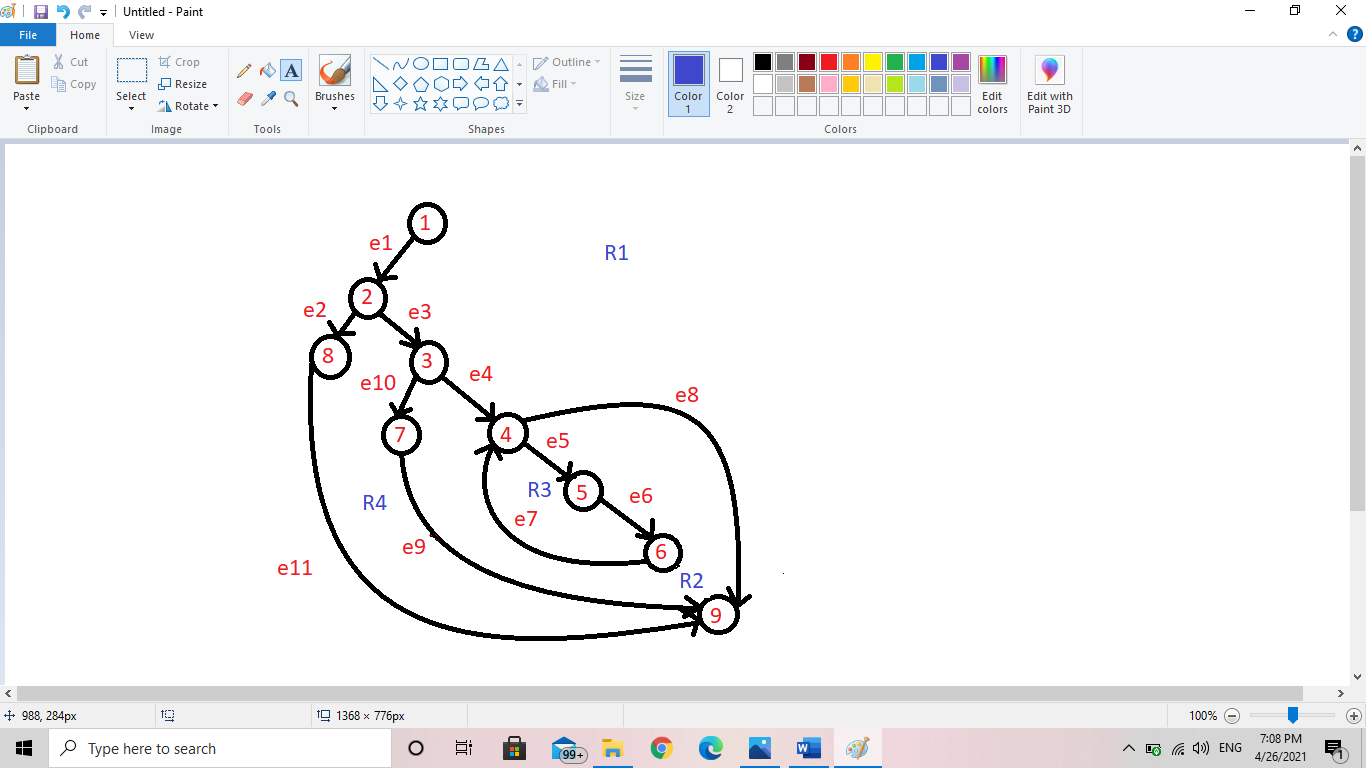
= $ 39K

1. **Testing**

**9.1. Pseudo Code for Searching Bus**



* 1. **Flow Graph**



* 1. **Calculating Cyclomatic Complexity**

Cyclomatic complexity is a software metric that provides a quantitative measure of the logical complexity of the program. When used in the context of the basis path testing method, the value computed for cyclomatic complexity defines the number of independent paths in the basis set of a program and provides the upper bound for the number of test cases that must be conducted to ensure that all statements are executed at least once.

**No. of Edges(E)=11**

**No. of Vertices(V)=9**

**No. Regions(R)=4**

**No. of Predicate Nodes(P)=3**

Predicate nodes are the nodes that have two or more edges emanating from them. Here predicate nodes are **2,3** and **4**.

1. **Using the number of regions:**

Cyclomatic Complexity =No. of Regions = R = 4

1. **Using formula E-V+2:**

Cyclomatic complexity = E-V+2 = 11-9+2 = 4

1. **Using formula P+1:**

Cyclomatic complexity =P+1 = 3+1 = 4

**INDEPENDENT PATH**

* **1-2-8-9**
* **1-2-3-7-9**
* **1-2-3-4-9**
* **1-2-3-4-5-6-4-9**

The number of paths depicts the maximum number of test cases required. In Basis Path Testing such test cases are prepared that will force execution of each path in the basis set. Here, number of independent paths are 4, which leads to the generation of 4 test cases.

* 1. **Test Cases**

|  |  |  |  |
| --- | --- | --- | --- |
| 1. Test Case No. | ROUTE NUMBER | LOCATION | Expected Output |
| 1. | **User** enters an invalid Route number: 420 | **OFF** | **Alert Message:**  The entered route number does not exist! and the application is not able to access the location.  Please try again later |
| ­2. | **User** enters a valid Route number: 118 | **OFF** | **Alert Message:**  The application is not able to access the location.  Please try again later |
| 3. | **User** enters an invalid Route number: 420 | **ON** | **Alert Message:**  The entered route number does not exist! |
| ­4. | **User** enters a valid Route number: 118 | **ON** | The application is navigated to the screen with the buses in that area. |

1. **Risk Analysis**

A risk is a potential problem which may or may not occur in the future. The risk table is used for risk projection/ estimation. It acts as a projection tool for risk managers to identify and understand different risks.

**10.1. Risk Table**

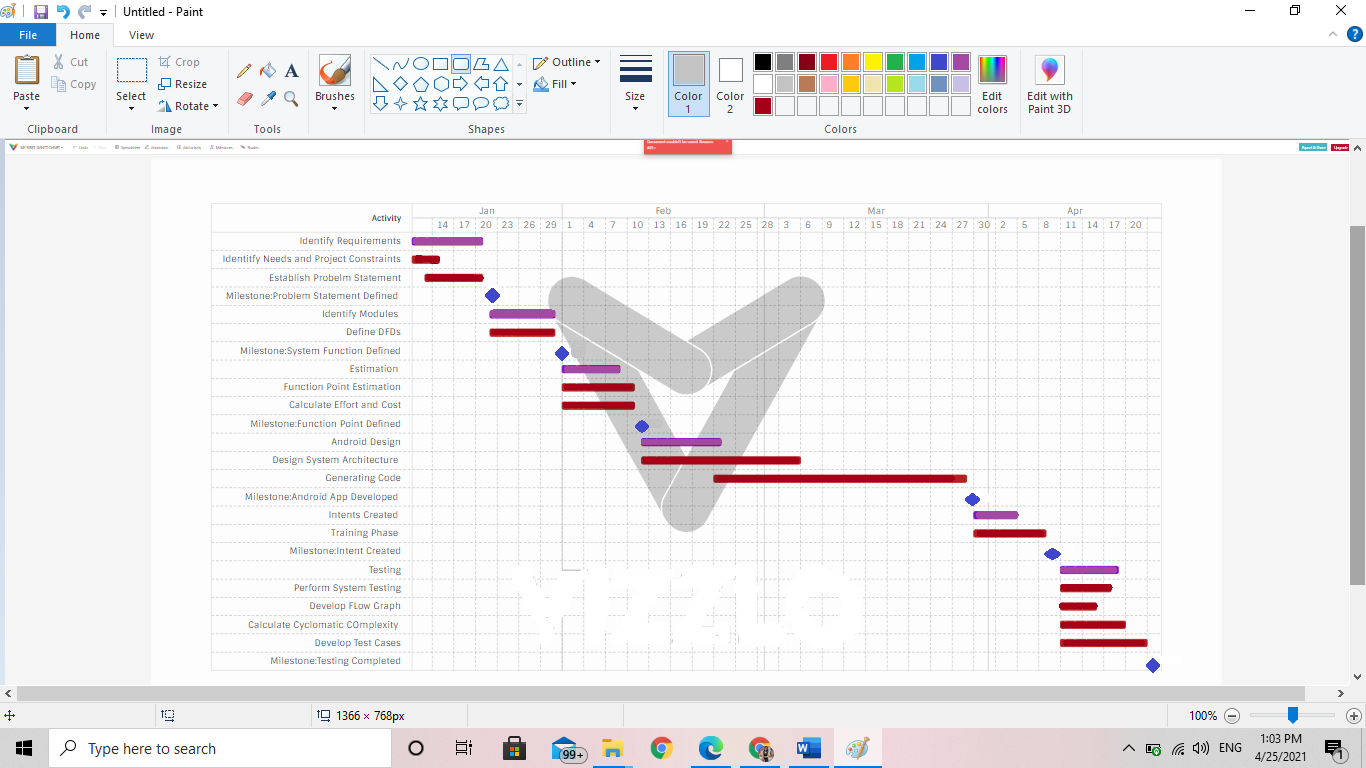
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Risks** | **Category** | **Probability** | **Impact** | **RMMM** |
| Size estimate may be significantly low. | PS | 60% | 2 |  |
| A larger number of users than planned | PS | 40% | 3 |  |
| Less use from planned | PS | 65% | 2 |  |
| End users resist system | BR | 40% | 3 |  |
| Technology will not meet the expectation | TE | 30% | 1 |  |
| Lack of training in tools | DE | 85% | 3 |  |
| Requirement understanding | PS | 50% | 2 |  |
| Inexperienced staff | ST | 65% | 2 |  |

ST: Staff Size or Experience DE: Development Environment

PS: Product Size BR: Business Risk

TE: Technological Environment

1. **Timeline Chart**



1. **Interfaces**
2. Home Screen



1. Login Slide

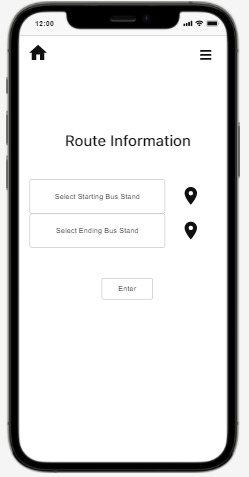
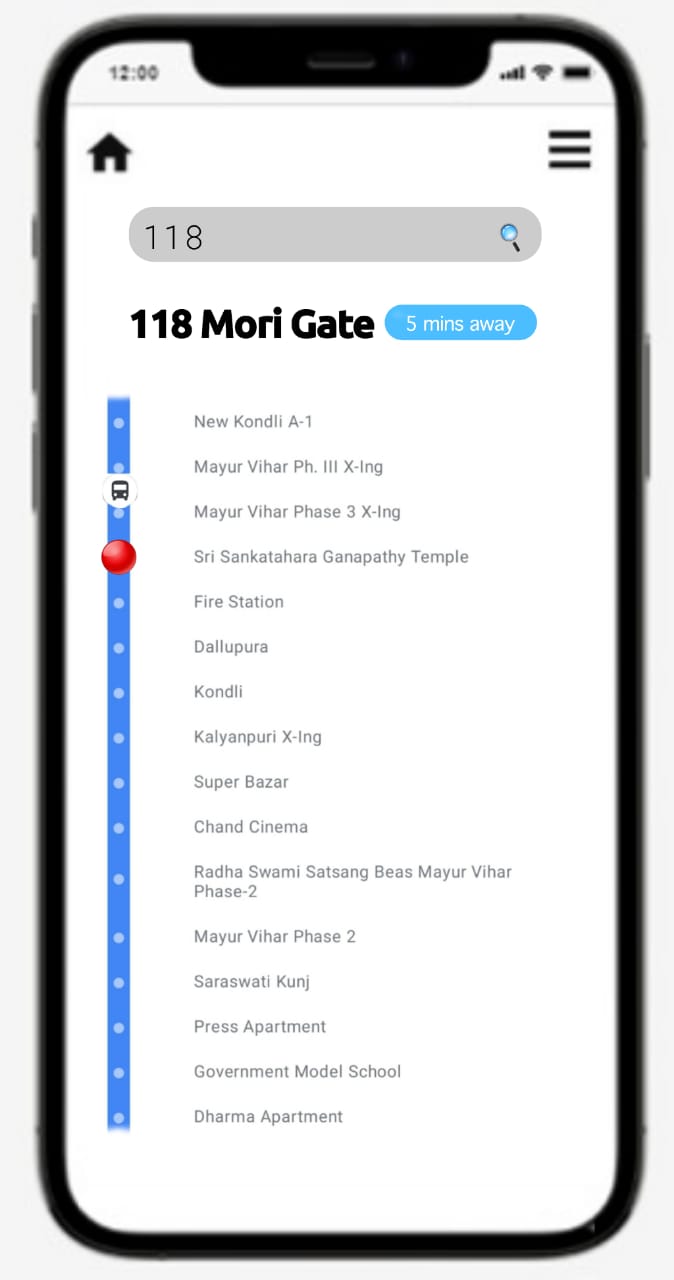


**12.1. Passenger Side**

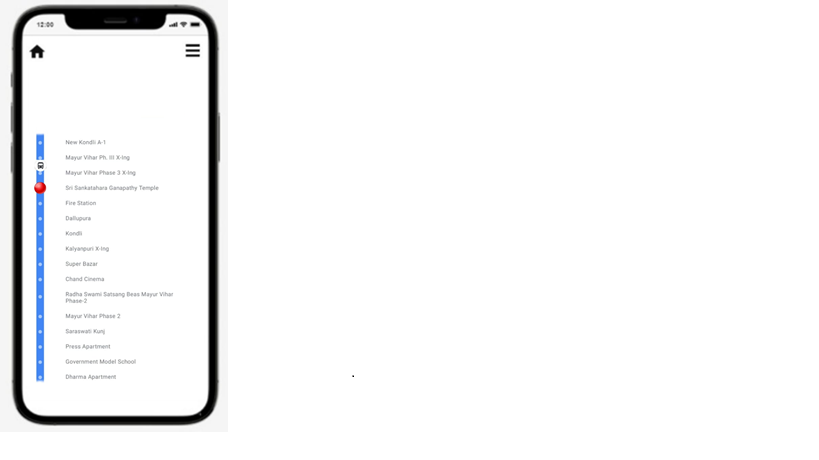
1. Content of the Apps



1. Route Between Bus Stands

1. Bus Route

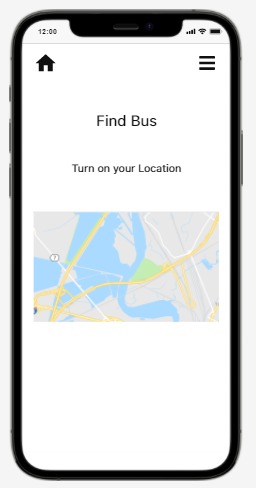


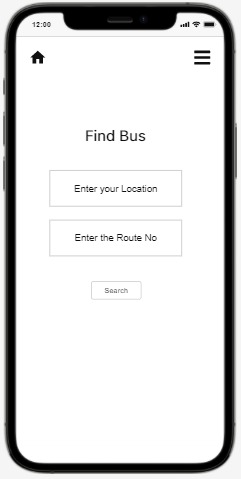
**DELHI BUS ROUTE**

1. DTC Bus Map



1. Bus Nearby



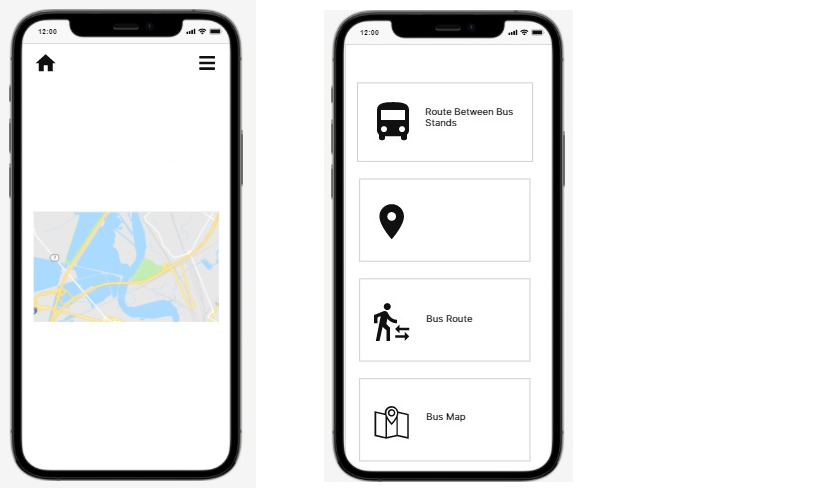
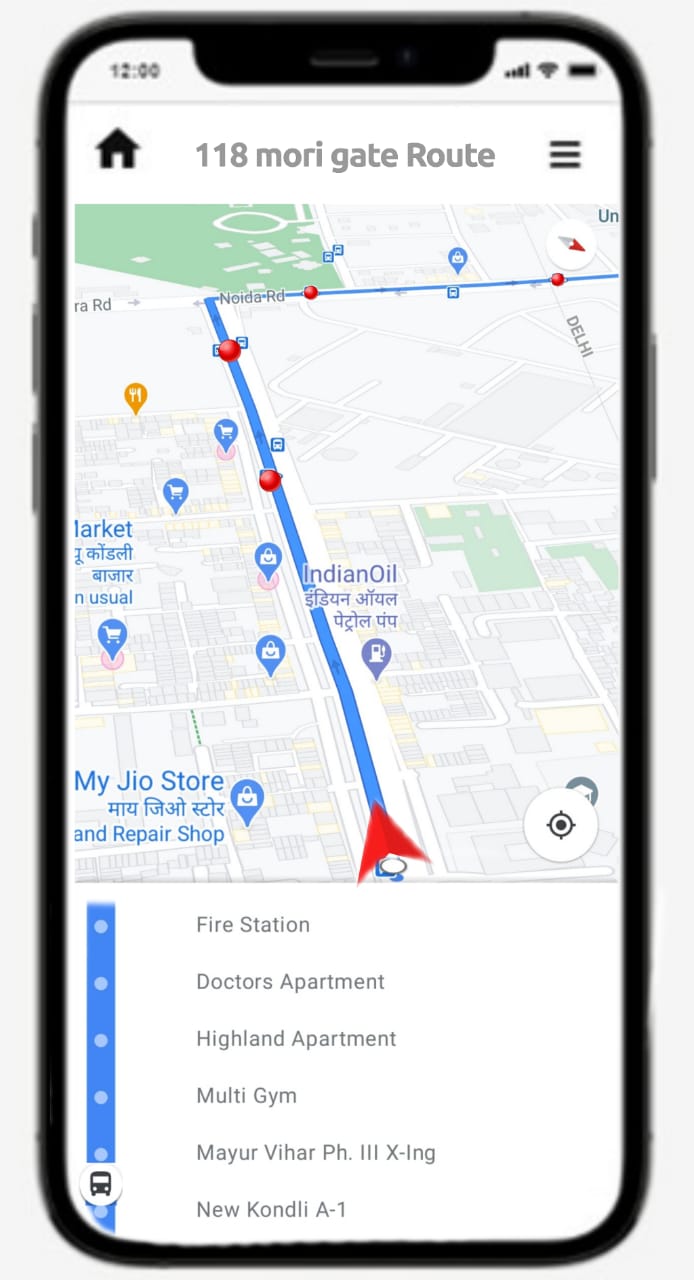
 

* 1. **Driver’s Side**

1. Content



1. Update location

TURN ON YOUR LOCATION

**UPADTE LOCATION**

1. **Bibliography**

Books:

1. Software Engineering: A Practitioner’s Approach: VII edition, Roger S. Pressman.
2. An Integrated Approach to Software Engineering, III edition, by Pankaj Jalote.