A picture containing text

Description automatically generated

System design

**Loza Tadesse, Amir Kemal, Nahome Hailemichael, Yanet Woldemichael**

Icon

Description automatically generated with medium confidence

Table of Contents

[**Executive Summary** 2](#_Toc88150543)

[**Design Strategy** 3](#_Toc88150544)

[**Method Specification** 5](#_Toc88150545)

[Method Specification Form 1 5](#_Toc88150546)

[Method Specification Form 2 6](#_Toc88150547)

[Method Specification Form 3 7](#_Toc88150548)

[Method Specification Form 4 8](#_Toc88150549)

[Method Specification Form 5 9](#_Toc88150550)

[**Relational Database** 10](#_Toc88150551)

[**UI Mockups** 11](#_Toc88150552)

[**Physical Architecture** 13](#_Toc88150553)

[Deployment diagram 14](#_Toc88150554)

# **Executive Summary**

For the Mini-Bus shuttle system, we broke up the proper way to design the system in three options: using an alternative diagram (with the alternatives being the choices to use packaged software), outsourcing, or designing the system in-house. Following this we then decided to use packaged software for its ease of access, familiarity, timeliness in terms of applying it to our system, and the fact that we do not have enough workers to design the system in house. Outsourcing comes up as a possible second option by taking in-house design out of consideration.

Our databases were designed to display an efficient route for the drivers and enable our consumers to track the path of the shuttles that are in transit. From there, our UI (User Interface) design for our system was established, allowing the consumers to track the shuttle to see the available stops and the destinations. The UI also depicts how the drivers will work depending on the information collected from students to see which routes will be used frequently as per the request of most consumers on the route and times needed.

Finally, the physical architecture for the system was designed, detailing our specific hardware and desktop setups. We will have an approximate of 1000 DSU students accessing our tracking system. We will provide a table for the drives with 8GB RAM, I5 processor and 15’’ screen. This is done so that they can see their route and track the van. As a medium of communication between the devices and server, we will have a management computer set up to create fluid communication.

# **Design Strategy**

The system we plan on building is not unique, and because it makes little sense to waste a lot of time for no reason, we plan to buy packaged software that has already been written rather than developing our own custom solution. There are many companies out there who use GPS to display accurate routes for both provider and consumer. Because we have a clear idea of the system, we want to build it faster to purchase, install, and use. The other thing that made us choose the package software is the fact that we can have a free trial of the packaged solution. This means we can get a hands-on test of the product you are looking at. This makes it easy to test the software to see if it is what you are looking for.

By using this strategy, we are still susceptible to the risks that come with it. While a packaged solution may be cheaper up front, it may end up costing us over time. This includes monthly costs, persistent fees, purchasing additional features, and the opportunity cost of not being able perform certain actions that you could with custom software. Another issue is that we cannot have any control over the software. You do not own the source code, nor do you have any say in how it is built. Updates and changes may be dropped on us that cause the software to no longer suit your needs. Most packaged applications allow customization, or the manipulation of system parameters to change the way certain features work. Which in some way will make it a bit flexible for us to customize the exact software that will work for us. Also, we can do a workaround which is a custom-built add-on program that interfaces with the packaged application to handle special needs. This is a subtle way to create needed functionality that does not exist in the software package. To determine what vendors to use, we will be using the RFI or request for information strategy. This will help us get a better idea of what each vendor has to offer and help in reducing our options down to the right vendor. We will be able to find out if a vendor has the skills and services, we need to complete this system.

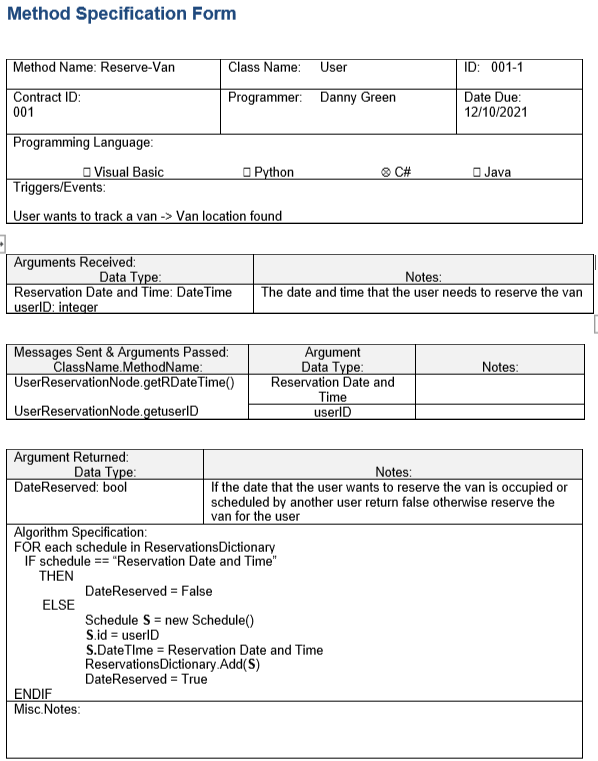
Alternative Matrix:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Evaluation**  **Criteria** | **Relative importance (weight)** | **Alternative1: packaged software** | **Score (1-5)** | **Weighted score** | **Alternative 2: Outsourcing the system** | **Score (1-5)** | **Weighted score** | **Alternative 3: custom development** | **Score (1-5)** | **Weighted**  **Score** |
| **Business need** | **20** | Business needs are common | **4** | **80** | Business needs are that important | **4** | **80** | The business needs are unique | **3** | **60** |
| **Project skill** | **10** | Small skills required | **4** | **40** | No skills required | **4** | **40** | most work is hands on | **2** | **20** |
| **Benefit** | **20** | Implementation is quick and easy does not require many people. | **3** | **60** | Small to no work required from our end. Also, it can be implemented efficiently. | **3** | **60** | The software is mostly built to the liking of the company by themselves | **4** | **80** |
| **Price** | **15** | 4500 | **4** | **60** | 7000 | **2** | **30** | 2500 | **3** | **45** |
| **Drawback** | **15** | There is some restriction in control of the software depending on how the software was designed for | **3** | **45** | The is no control of the software | **2** | **30** | The is total control of the software | **4** | **60** |
| **Reliability** | **10** | With the right provides can acquire reliable software | **3** | **30** | With the right requirements provided can be reliable with the right company | **4** | **40** | Depends on the skills of the developers in the company | **2** | **20** |
| **Time frame** | **10** | Implementation in significantly less | **4** | **40** | Implementation in moderate times | **3** | **30** | Implementation is slow | **2** | **20** |
| **Total** | **100** |  |  | **355** |  |  | **310** |  |  | **305** |

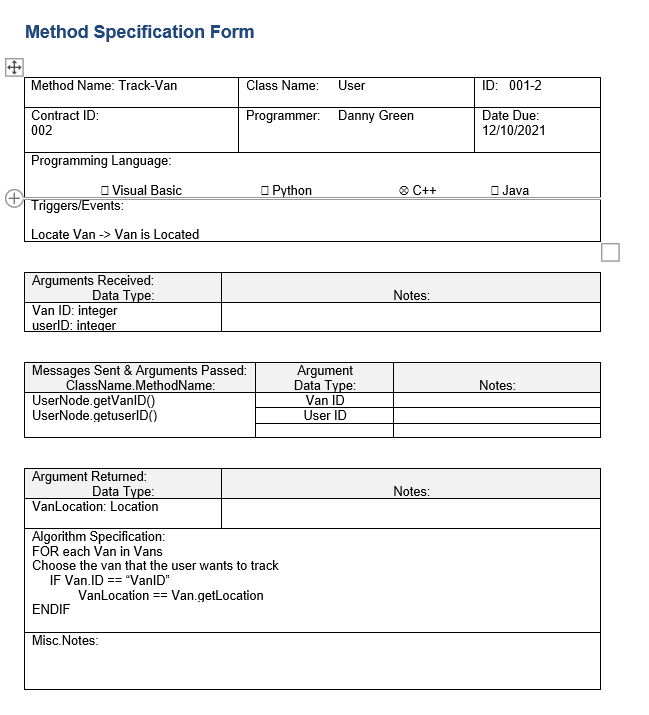
We have decided to get packaged software. We will then send an RFI to vendors, which will help us in determining which vendor can give us the best deal on the software. Then we can purchase our software and implement it.

# **Method Specification**

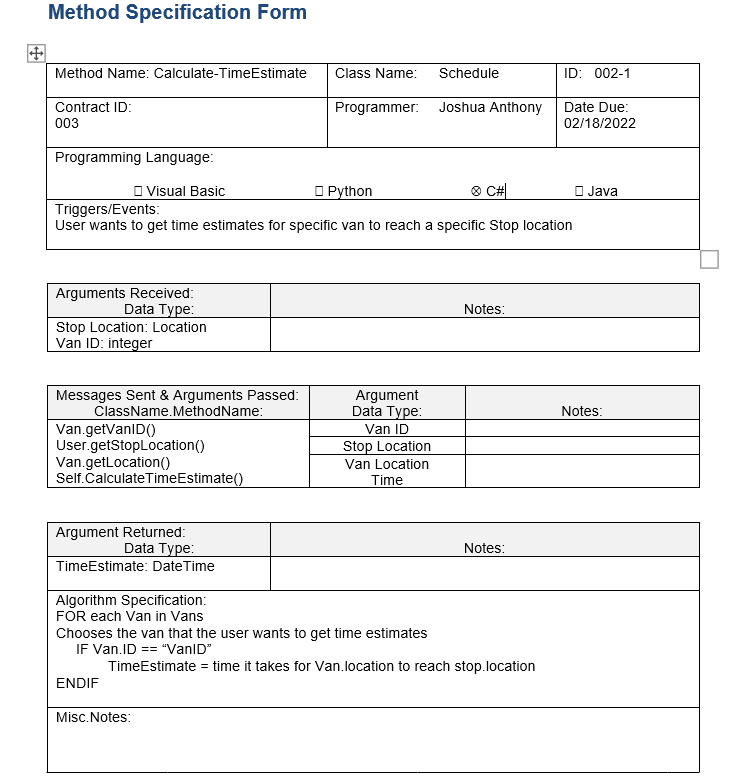
### Method Specification Form 1



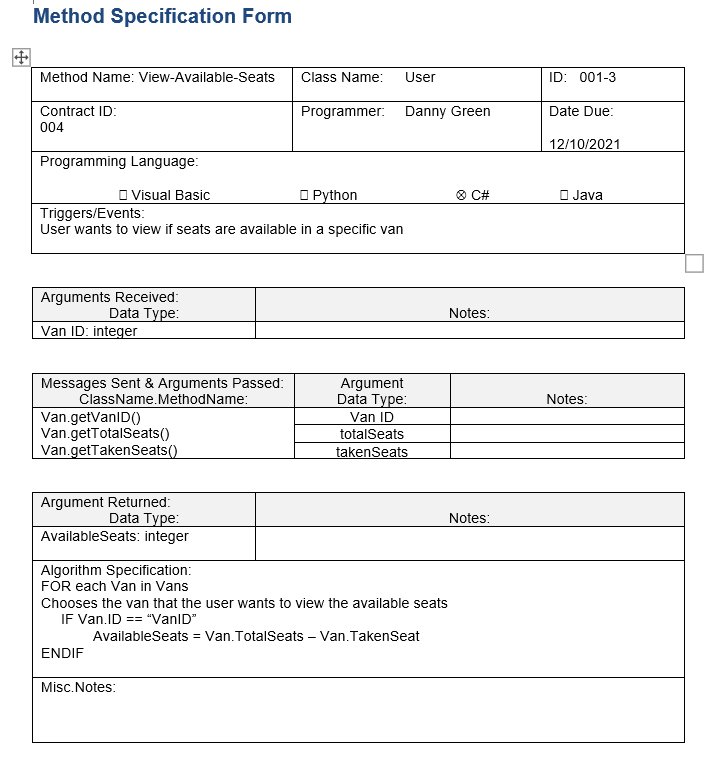
### Method Specification Form 2



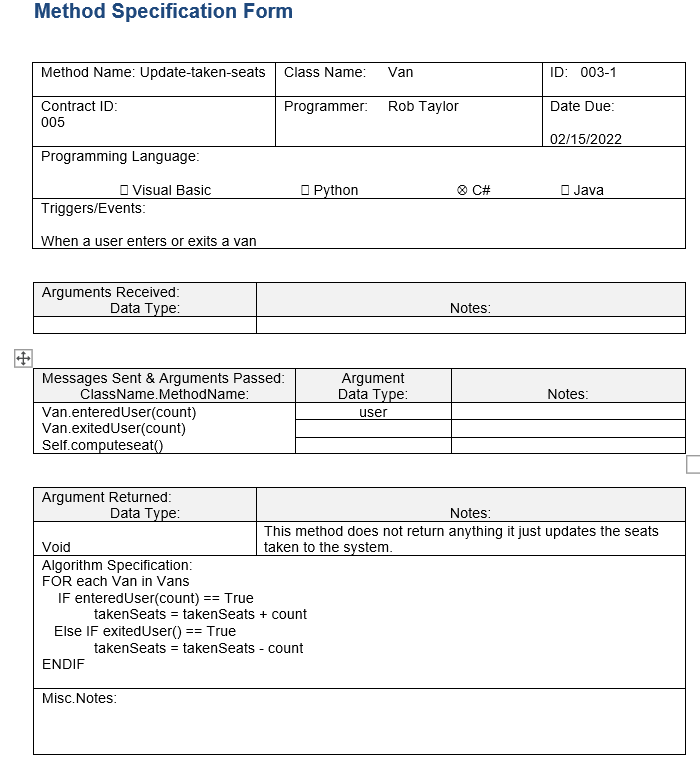
### Method Specification Form 3



### Method Specification Form 4



### Method Specification Form 5



# **Relational Database**

Graphical user interface, application

Description automatically generated

# **UI Mockups**

View schedule

Graphical user interface, application

Description automatically generated

Reserve van

Graphical user interface, application

Description automatically generated

View Route

A picture containing diagram

Description automatically generated

Track van

Graphical user interface, application

Description automatically generated

# **Physical Architecture**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Specification | Client server | Standard Web Server | Standard Application Server | Standard Database Server |
| Operating System | IOS, Android,  MacOS  Windows 10 | Linux | Linux | Linux |
| Special Software | Google maps | Apache | Apache | Oracle |
| Hardware | Desktop Computers  8GB RAM  250 GB Disk Drive    I5 processors  2-22’’ monitors    Tablets:  8GB RAM    I5 processor  15’’ | 16 GB Memory  1 TB HDD    Intel Xenon (server processor I7)    Virtualized (no monitor) | 32 GB Memory  2-1 TB Disk Drives    Intel Xenon e5-2600 | 32 GB  4-1 TB Hot plug disk drives intel xenon e5-2600    Virtualized (no monitors)    17’’ monitor  1 mouse  1 keyboard |
| Network | Gigabit ethernet  100Mbps  Wireless | Gigabit ethernet  3000Mbps  Wireless | Gigabit ethernet  3000Mbps  Wireless | Gigabit ethernet  3000Mbps  Wireless |
| Warranties | Give warranty for a period of 12 months from the date of purchase of GPS | 5-year licensing when bought as a package through Microsoft computer vendor does 5-year | 5-year licensing when bought as a package through Microsoft computer vendor does 5-year | 5-year licensing when bought as a package through Microsoft computer vendor does 5-year |

### Deployment diagram

Timeline

Description automatically generated