Data Management Plan

**Types of data produced:**

Our project is all about tracking the attendance of the student, we are required to generate the dynamic data from the student by scanning the QR code. By scanning the QR code, the details of the student should match the details present in the static database and mark the attendance. Static database of all the users is required and will be created for running the simulation.

As the project progress, we are preparing the data about stand-up meetings, client meetings, team meetings, weekly iteration plans, team status reports, project management plan, requirements document, use case diagrams, specifications, issues tracking document, data management plan, database design, test plan, test suite, project code, user manual and installation manual.

**How the data will be created?**

Note taker is responsible for saving the data in the client meetings, stand up meetings and also the team meetings assigned by the project team. Iteration plan is created at the starting of the project and then as a team we followed accordingly and submitted all of our team work on Fridays. Project management plan, requirements document, use case diagrams, specifications, issues tracking document, data management plan, database design, test plan, test suite, project code, user manual and installation manual are discussed by the project team and completed the work collaboratively.

Data that is required for our project database can be collected in our University. This data includes details of Students, instructors, departments and interactions between them. Remaining data that is required for project database are collected from the client meetings.

Simulation will produce live data of trains which includes the routes of trains, current speed and current location. It will produce live track data which includes sections locked by trains, sections available and directions of switches.

**Storage of data:**

Documentation and coding of the Data that are created by our team will be stored in GitHub repository created by our team named Student-Attendance-Tracking. Format that the documentation data stored is .docx and .txt in order to make it more accessible. Image data is stored in .jpeg, .png or .gif formats. Format for coding the data is stored in respective format that our team is going to decide. The data that we created are stored in directory of Student-Attendance-Tracking. Documentation and code is stored in subdirectory. This include Meetings, weekly iteration plans, team status report, project management documents, requirements document, database design, test plan, test suite, user manual and installation manual. The code subdirectory will include the code of the project and resource files necessary for the project to work.

**Addressing the Security for the data used:**

Security is the key element for any project to work successfully for any user to trust the project. The security can be bleached in multiple ways for any online project such as the SQL injection attacks, database intrusion attacks and even by getting the session id for capturing the session cookies and capturing the data of the user. These are some of the main attacks which may lead to the problems of security attacks.

There are some of the key features which allows our system to be safe and secure for any user to user it.

**Express Session Package:** This is an NPM package which mainly **e**ncrypts user data for that particular session which states that the details for that particular session such as the session id is not stored in the client side it is stored in the server side which reduces the possibility of retrieving the cookies for that particular session. This package does not all the user to re initiate the session id, this can be done by changing the option to view the session id as false.

**Sanitizer Package:** This is an NPM packagewhich is used against the SQL injection attacks**.** Using this package the user would be able to enter any test data into the fields but cannot enter any type of SQL commands. This feature encrypts all the input commands and allows only those text with no sql commands in it.

**Parse Database:**  Parse database follows the ACL based permissions in accessing the data in the database. Using this database a user has the ability to create the database with different Class-Level Permissions which are used for authenticating the users based on the type of user. In this we will the using the commands such as “requiresAuthentication” which can be set to true or false. This database can also be used for including the level of transparency, such as the user can hide the passwords of each and every used registered in to the project.

These are some of the security features involved in the project to ensure privacy of the user. There are also many other validations for all the input fields which allows the user to have a next level of safety.

**Archiving and preservation of data:**

Data will be preserved in GitHub repository after completion of the project which is a stable resource for making data readily available. Data will be made available indefinitely on GitHub. This data will include all of the documentation and code data.

**ER Diagram**

