Software Requirements Specification

For

Student Engagement Tracking System

Adam Abdellaoui

Tony Andrews  
Philip Fernandez

Austin Lambeth

Bronson Lane

Anthony Shedlowski

September 19, 2018

**Table of Contents**

1. **Introduction……………………..……………………………………………….. 3**
   1. Purpose………...………...………...………...………...………...…….... 3
   2. Project Overview………...………...………...………...………...……….3
   3. Tobii Eye Trackers………...………...………...………...………...…….. 3
2. **Requirements………...………...………...………...………...………...…….....3**
   1. System Server………...………...………...………...………...………….3
   2. Teacher………...………...………...………...………...………...………. 4
   3. Student………...………...………...………...………...………...………..5
   4. Database………...………...………...………...………...………...…….. 5
   5. Administration………...………...………...………...………...…………. 6
   6. Login………...………...………...………...………...………...…………. 6
3. **Use Cases………...………...………...………...………...………...…………... 6**
   1. Server………………………………………………………………………6
   2. Student………...………...………...………...………...………...………. 7
   3. Teacher………...………...………...………...………...………...………. 7
   4. Admin………...………...………...………...………...………...………… 8
4. **Design Constraints………...……….………...………...………...………...…. 8**
   1. Performance Requirements………...………...………...………...……. 8
   2. Assumptions………...………...………...………...………...…………... 9
5. **Dashboard Mockup……………………………………………………………..10**
6. **Logical Diagram…...……..………...…………..............................................10**
7. **Database Diagram……………………..…...………...………..………...…….. 11**
8. **Use Case Diagram………………………………………...……………………. 11**
9. **Introduction**
   1. **Purpose**

This software requirements document describes the functional and nonfunctional requirement for the Tobii Eye Tracking system. This document is for use of members of the team and to verify the functionality of the software.

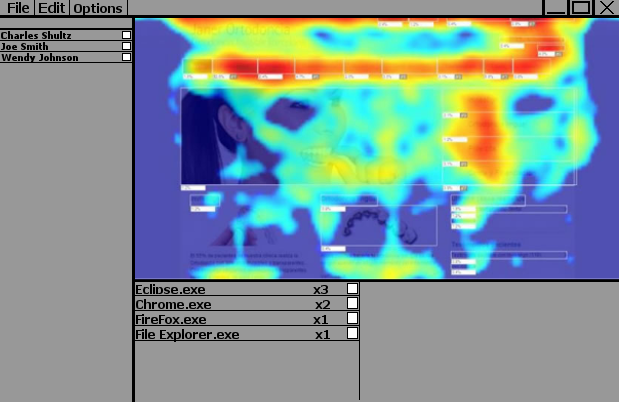
* 1. **Project Overview**

This project has a goal of streaming data from multiple Tobii Eye Trackers. This data will be stored data in a database and streamed to a display in a teacher application. This data will include where the student is currently looking and other data such as currently opened windows.This should allow for the teacher to better determine if their students are currently engaged in the lecture or not.

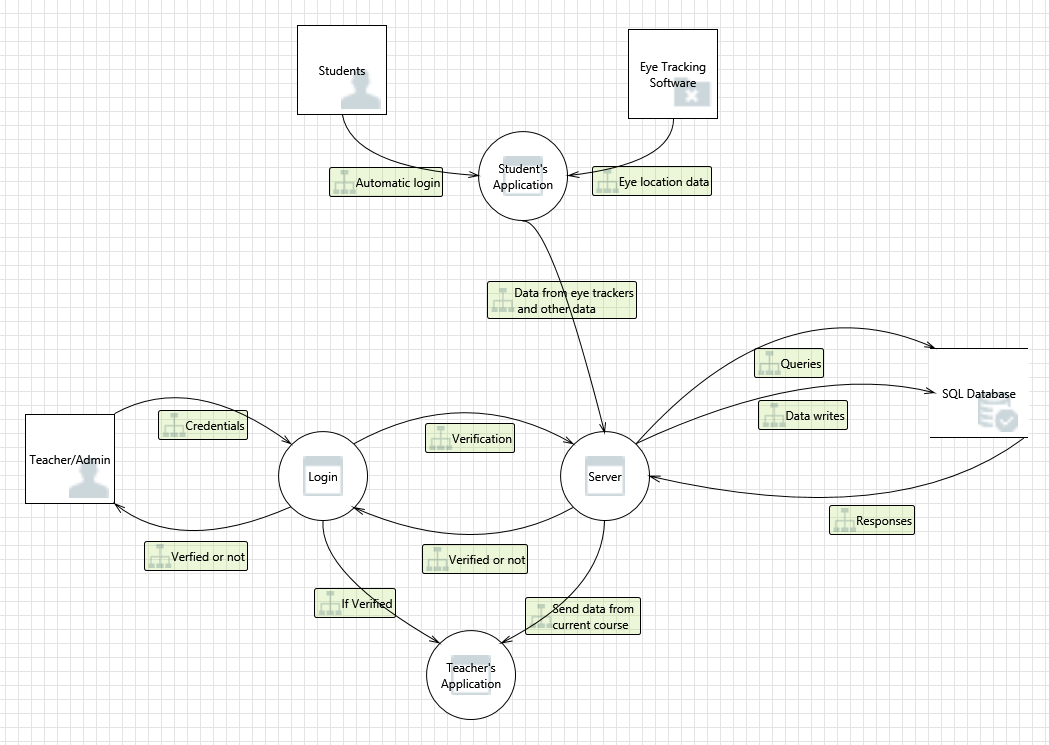
* 1. **Tobii Eye Trackers**

The eye trackers are hardware designed by Tobii Technology. They use cameras, illuminators, and algorithms to track eye movement across the computer monitor.

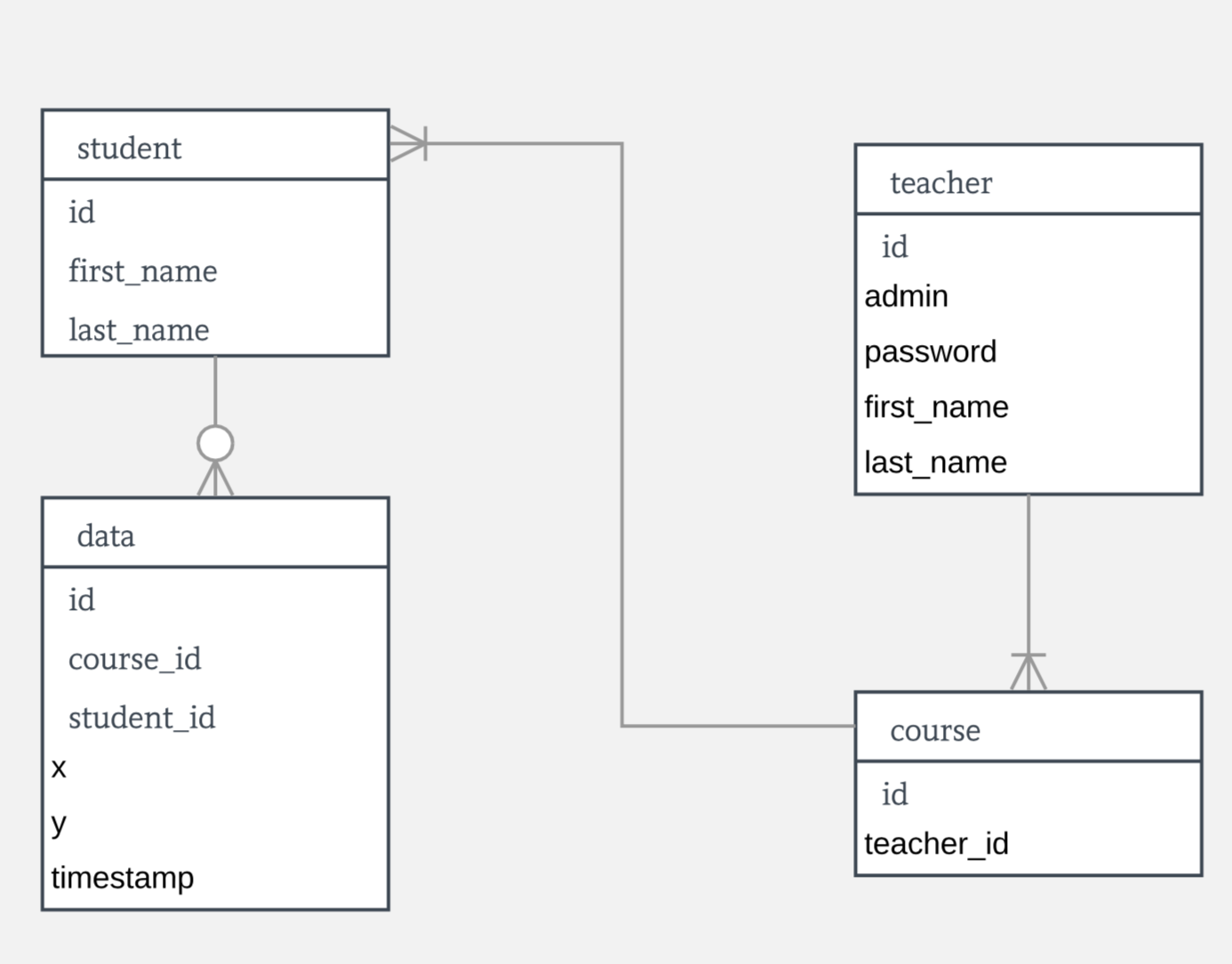
1. **Requirements**
   1. **System Server**
      1. The system shall be able to stream information in real-time from the clients to the server.
      2. The system shall be able to connect one or more students at the same time.
      3. The system shall send information from the student’s computer to the server
      4. This information should be the student’s ID, where they are looking, time, and others.
   2. **Teacher**
      1. The system shall display the teacher’s graphical user interface.
      2. The teacher shall be allowed to record new data and review already recorded data.
      3. The system shall allow for the teacher to review data based on specific students, courses, and/or time.
      4. The system shall allow the teacher to select a course to view the students actively displayed on the GUI.
      5. The teacher’s GUI shall show the currently chosen filter to the teacher.
      6. The system shall have the ability to swap filters.
      7. The system shall show the students that are currently connected to the server.
      8. The system shall be able to select one or more students to highlight on the aggregated heat-map.
      9. The system shall show applications that are currently open on students’ computers.
      10. The system shall allow the teacher to start and stop recording data as needed.
      11. The teacher should be able to choose an area of the screen and capture attention data in that specific area.
      12. The teacher shall be able to choose the class that they want to capture or view data from.
   3. **Student**
      1. The system shall automatically connect the student to the server.
      2. The system shall be able to automatically retrieve student’s username when they log in.
      3. The system shall send eye tracking data.
      4. The system shall send other data, such as current open programs.
   4. **Database**
      1. The system shall automatically store data as it is received.
      2. The system shall be able to retrieve data from the database.
      3. The system shall be able to use the retrieved data for retroactive analysis.
      4. The retroactive analysis shall be able to be displayed in the teacher’s GUI and with their chosen filter.
      5. The system shall store the teacher’s username and passwords for verification.
      6. The database shall require authentication to access it, to help ensure privacy is kept.
   5. **Administration**
      1. The system shall give verified administrators the ability to alter the database.
      2. The system shall let administrators add and remove teachers, students, and courses as changes are needed.
      3. The system shall allow administrators the ability to reset users’ passwords.
   6. **Login**
      1. The system shall allow users the ability to sign in by using their username and password.
2. **Use Cases**
   1. Server
      1. The server controls the information coming from the students and going to the teachers.
      2. The server receives the tracked eye data and other information from the students.
      3. The server sends all of the data from the students and saves them to a database to be retrieved.
      4. The sends the data from the students to the the teacher per request.
   2. Student
      1. The student uses their school login to log in to the computer. The software will pull the student’s username from the submitted login information.
      2. The software automatically starts and tries to connect to the server.
      3. After connecting, the system will start sending data to the server when requested.
   3. Teacher
      1. The teacher enters their username and password to log in to the application.
      2. The teacher can select a class to view from a list of the classes they currently teach.
      3. The teacher can choose between recording for a class or reviewing past recordings.
      4. The application will retrieve the requested information from the server.
      5. The teacher can select a filter for the data.
      6. The teacher can select from a list of students to see individual data.
   4. Admin
      1. The admin enters their username and password to log in to the application.
      2. The admin is authenticated and allowed access to the database.
      3. The admin can add or delete courses from a course list.
      4. The admin can add or delete users from a user list.
      5. The admin manages the database connection.
      6. The admin can reset a teacher’s password.
3. **Design constraints**
   1. Set up in a classroom environment
   2. Easy and Simple to set up
   3. Store historical data
   4. Stream processing parallel process and data showing 2 - 3 things at once on the dashboard
   5. Scroll back in time and play Heat map which cursor showing the percentage
   6. Performance Requirements
      1. The system shall accommodate 30 users at once per classroom.
      2. The system should accommodate as many users as the system can computationally handle.
      3. The system shall be able to display the data collected from the eye tracker with a 1 second delay or less from the moment it is captured to when it is displayed on the monitor.
      4. The system shall allow scrubbing of the historical data with less than 1 second of delay in when the time is picked to when it is displayed.
   7. Assumptions
      1. The calibration of the eye trackers will be handled by a seperate program that has calibration, using calibration dots that scans the position of the eye pupils.
      2. Students that log on are part of a database of users that the system is able to access.
      3. All students using the system have accepted that they are being tracked.
4. **Teacher Dashboard Mockup**



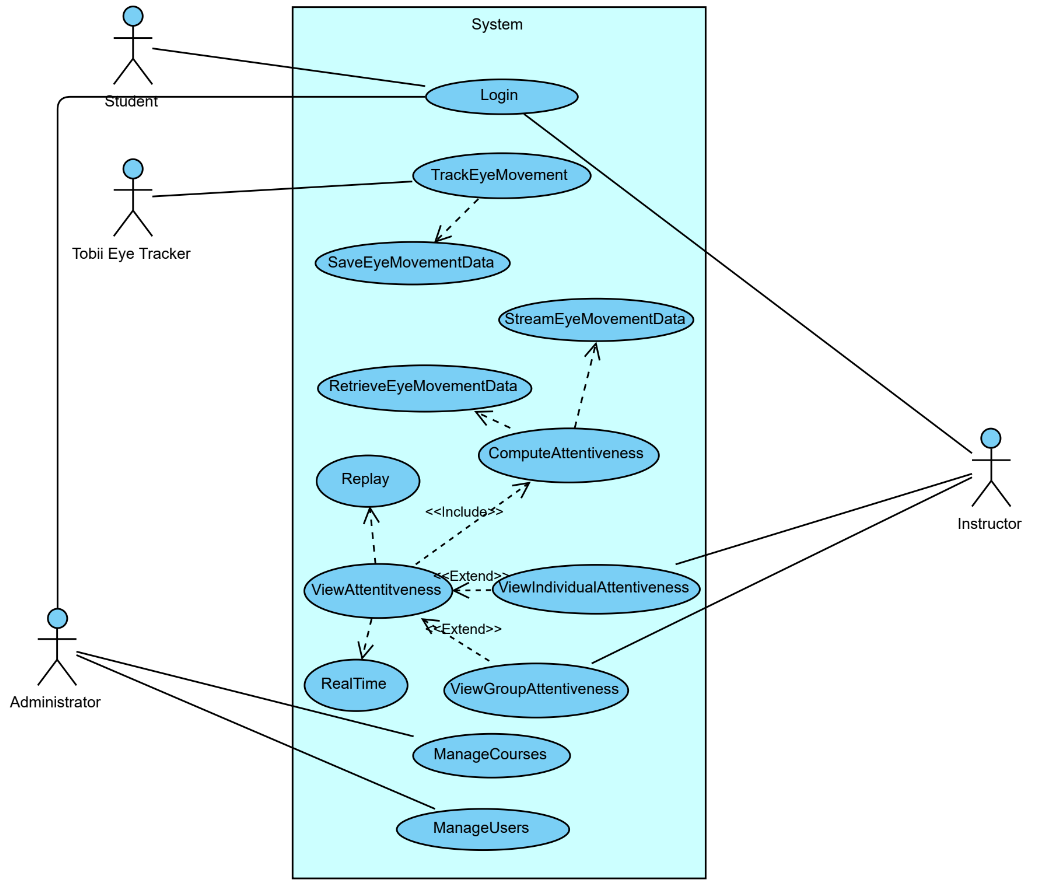
1. **Logical Diagram**

****

1. **Database ER Diagram**



1. **Use Case Diagram**

****