

**MACROHARD**

System Design Document

Team 1

ERYK CYGIELSKI

ANDREA BIALKOWSKI

NERMIN DEDOVIC

RANDY LY

JITEN PATEL

RAMEEZ ANWAR

DIMITRI VITYK

10/13/2020

## Revision History

|  |  |  |
| --- | --- | --- |
| Revision Number | Date | Comment |
| 1.0 | October 13, 2020 | Original Document |
|  |  |  |
|  |  |  |
|  |  |  |

## Table of Contents

Revision History …..……………………………….………………………………… 2

Table of Contents ….………………………….……………………………………… 3

Document Overview ….……………………………………………………………… 4

System Overview …..………………………………………………………………… 5-7

Hardware Design …..………………………………………………………………… 8

Software Design ……………………………………………………………………… 9

Database Design …....………………………………………………………………… 10

Glossary …………….………………………………………………………………… 11

## Document Overview

This report is a technical blueprint for the MacroHard.

This document has been developed by Eryk Cygielski, Andrea Bialkowski, Nermin Dedovic, Randy Ly, Jiten Patel, Rameez Anwar and Dimitri Vityk in the name of MacroHard for students, professors, and administrative faculty.

### Scope

The system will have student and faculty accounts to use the system at different capacities. There will also be administrator accounts in case of required administrative changes.

### Audience

The intended audience for MacroHard is school systems, students, and faculty in need of project management software. The schools could introduce the application to teachers who can have their students use it as a tool in group projects. The teachers could use this application to simplify the peer evaluation process as they can see all the work done by each student, and all reviews of that work. This would make grading easier than it currently is. MacroHard would make it easier to track student participation and involvement data.

## System Overview

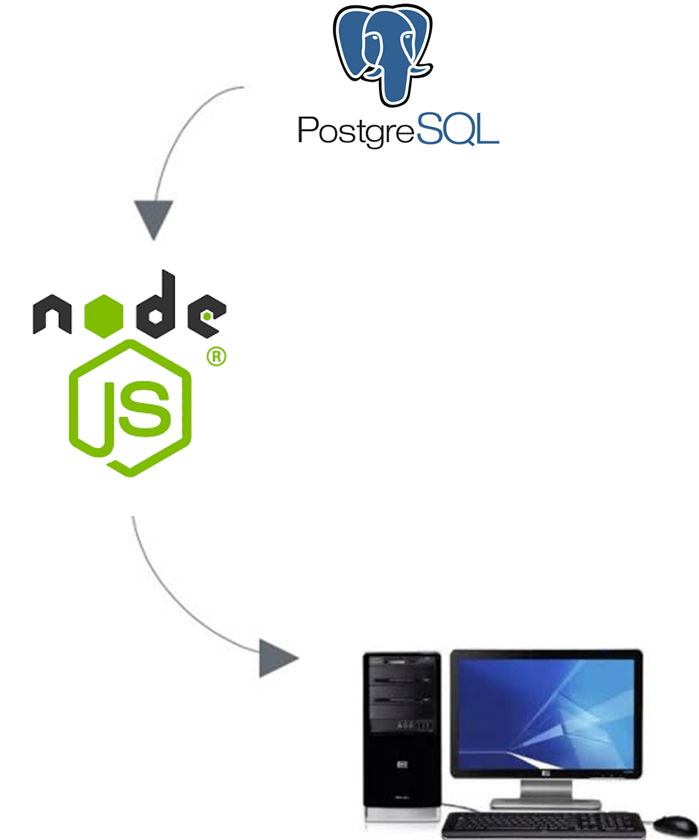
### Description

The proposed system for MacroHard is to incorporate various tools to produce a successful platform. The database will be created on PostgreSQL and the backend will be coded in Node.js, using express. The front-end will be created using HTML, CSS and JavaScript and it will all be hosted on a local machine. An overview of the data flow will be the Database to Node.js to the local machine and then to the User creating a simple sequence.

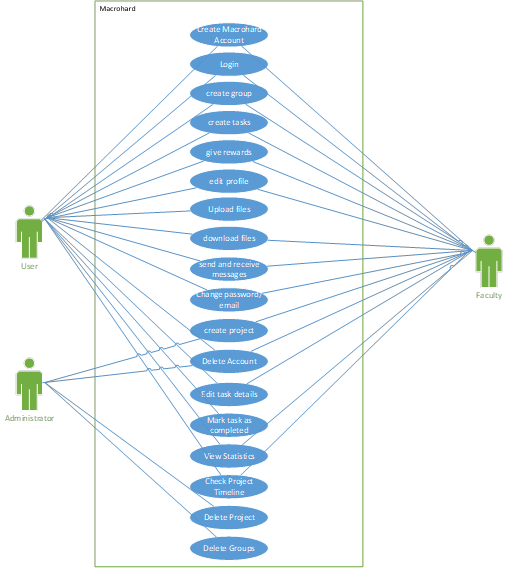
### System Architecture

Data flow:

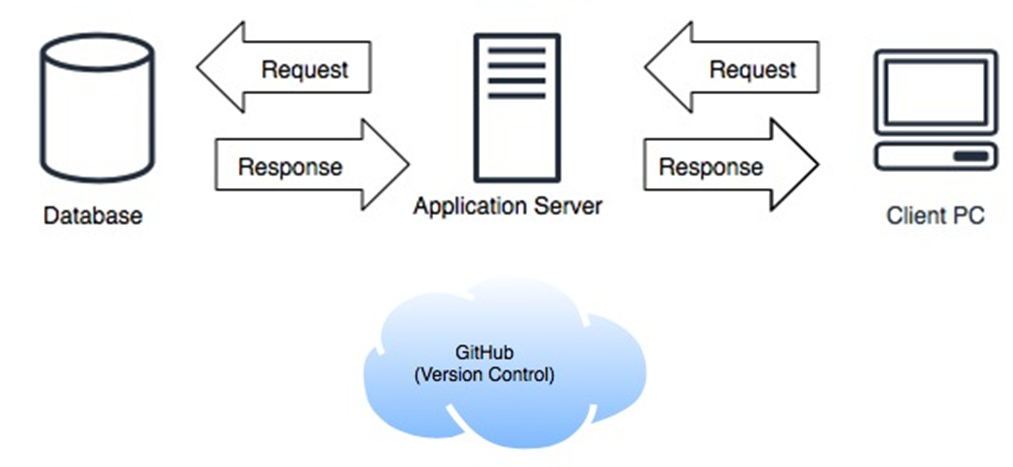
Test Data → Database → Node.js → Local Machine → User



### System Use Case Model



### System Design



## Hardware Design

### Overview

MacroHard is primarily designed as a web application, the hardware related to MacroHard is very limited. As referenced in Figure 1-3, the hardware used will not have any relation to MacroHard as all internet accessible client machines would reach the web application. In terms of hardware, only a local machine with the database and code is required, along with a mouse and keyboard for navigation and input.

## Software Design

### Overview

MacroHard uses third-party services to create the best and simplest usability for users. The various components of the MacroHard software application are explained below.

### Languages

**Backend:** Node.js   
**Frontend:** HTML / CSS / JavaScript  
**Configuration:** Express  
**Database:** PostgreSQL

### Reliability

Node.js is an open-source, cross-platform, back-end, JavaScript runtime environment that executes JavaScript code outside a web browser. Node is useful for developing applications that require a persistent connection from the browser to the server and is often used for real-time applications such as chat, news feeds and web push notifications. Express is a minimal and flexible Node.js web application framework that provides a robust set of features for web and mobile applications. Express provides a thin layer of fundamental web application features, without obscuring Node.js features. The database will be hosted on a local machine.

### Software Integration

The database is configured in PostgreSQL, which is connected and configured with Sequelize ORM, which maps our data into our application then coded through Node.js to enter and pull the necessary information. The information is then transferred to the client through PostgreSQL interface coding. The systems integrate together in order to provide the user timely information and responses.

## Database Design

Data Flow and Database Design Diagram

Graphical user interface, diagram

Description automatically generated

## Glossary / Terminology

Not Applicable