

Jordon Behr, Drew Bhavsar, Kevin  
DiBiase, Sarim Hussain, Rahul Saxena,  
Ronisha White, and Kevin Zhen

# ChalkBoard

## Table of Contents

Requirements Documentation.....	2
Design Documentation.....	3
- Overview.....	3-4
- Technical Requirements .....	4
- Tech Stack Visualization.....	4
- E-R Diagram.....	5
Product Backlog .....	6
Project Plan.....	7
- Sprint Chart.....	7-8
- Scrum Chart.....	8
Code Repository.....	9
Testing Plans and Results .....	10-11
Deployment Plan .....	12
- Scripts.....	12
- Deployment Plan.....	12
- Rollback Plan .....	13
Maintenance Plan.....	14

# Requirements Documentation

## Overview

In this project the team is designing a quiz application supporting both multiple choice as well as true and false questions. The app will include two types of views; Administrator and User. The roles separate privileges, allowing administrators to be able to create quizzes/questions, add new accounts, and administer the quizzes to specific users. Users will be able to take the quizzes and view their grades. The front end side of the application is made with a pleasant experience in mind and must place ease of use above all else.

## Functional Requirements

### Primary:

- Users can create an account and have access to quizzes that they can take.
- Administrators can create user accounts and quizzes for users to take.
- Administrators can create questions for quizzes, and store quizzes in a quiz bank.
- The sign-in (log-in) page can distinguish between users and admins.

### Secondary:

- Administrators can set a timer for the quiz.
- Administrators can set point values for answer choices with values of at least 0 and up to 5 points.
- Administrators can set the quiz to be available until a specific date.
- Administrators can delete old quizzes.
- Users and Administrators can view the quiz results.

# Design Documentation

## Overview

The team implemented Agile Scrum Methodology to help each member keep up with their workload and project goals, with the project design being dependent on integration. The three main components that were required to complete the project were the front-end, back-end, and the database. The languages used to develop the front-end were HTML and CSS with Bootstrap, a CSS framework, being used to aid the design of the front-end requirements. The Back-End of the project was developed using JavaScript in a Node.js runtime environment.

The main objective of the project is to allow administrators to create quizzes, allow users to take quizzes, and allow both to view the results of quizzes. A backlog was created to divide out the work for the team into weekly sections so that work would be completed in a timely manner. The development was able to pass the milestones of creating and implementing a UI design and building the functionality of the back-end using Node.js.

## Frontend

The team's frontend was designed by using Bootstrap, a CSS library. EJS was also utilized as a way to generate HTML markup with Javascript. The .ejs files connect Javascript with node\_modules from node.js, as well as allowing for the user to enter data into the MySQL database by using REST HTTP requests.

## Backend

The backend was made utilizing Express and node.js. Node.js was used to connect to the SQL database. These node.js files do everything from populating the database with various types of information to relaying HTTP requests to the frontend.

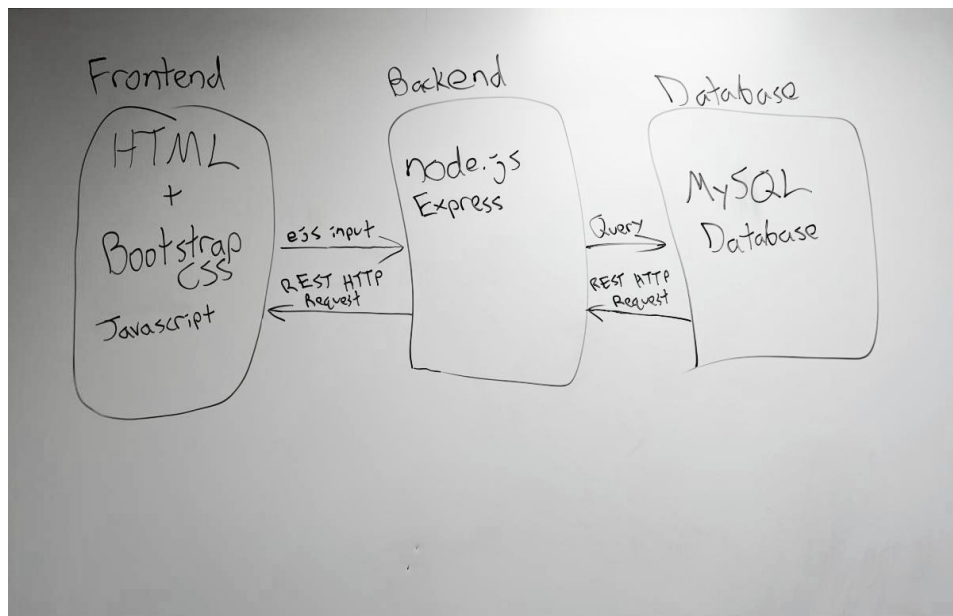
## Database

The database was created by implementing MySQL on the Microsoft Azure cloud platform. In this implementation, the team set up measures to distinguish between administrators and regular users. This enabled it to be queried and altered at any time remotely by users of the app. The team implemented multiple tables to represent users, questions, and the quizzes themselves.

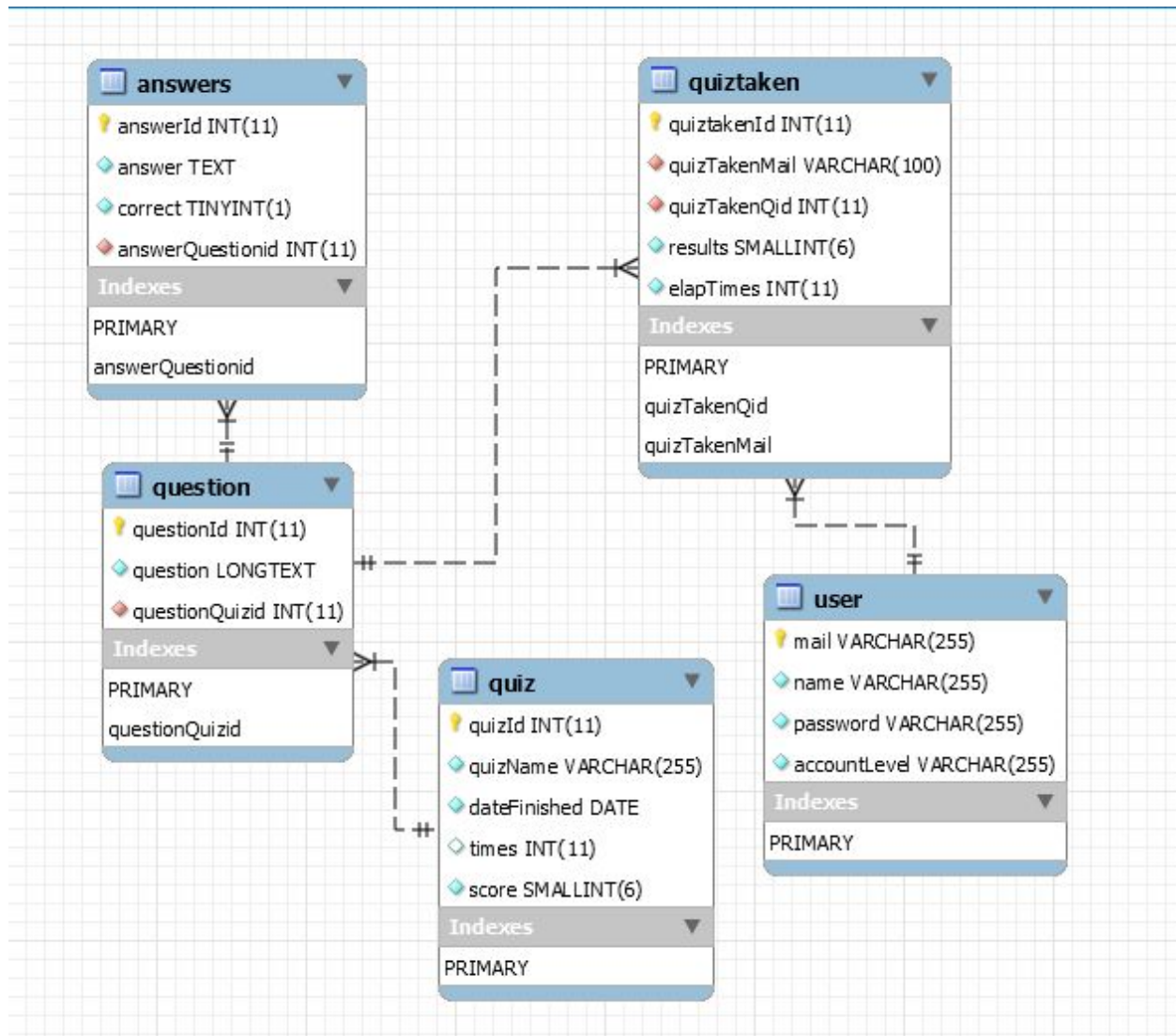
## Technical Requirements

- Set up a database with Azure connecting it with a localhost using MySQL database.
- Used bootstrap for the front-end UI design along with HTML.
- Utilized JavaScript in a Node.js environment for the backend.
- Have an API with an acceptable level of documentation.
- Deploy scripts for the database.
- Set the application live.

### Tech Stack Visualization:



**E-R Diagram:** Shown below is an Entity Relationship diagram that represents our MySQL backend.



# Product Backlog

## SCRUM Product Backlog

Story
As a user, I want to be able to take a test.
As an administrator, I want to be able to give a test to specific users.
As an administrator, I want to create a test with questions and corresponding answers.
As an administrator, I want to add or change questions and answers.
As an administrator, I want to use both multiple choice and true/false formats.
As a user and administrator, I want to login with a unique user id and password.
As an administrator, I want to be able to manage and create accounts.
As a user, I want to be able to select a test from a test bank to take
As a user, I should be able to see the result of my own test in hyperlink format.
As an administrator, I want to be able to view a listing of tests taken by a user.
As an administrator I want to have the results of all user's tests in hyperlink form.
As an administrator, I want to be able to provide feedback to users on their tests
As a user, I want to be able to view feedback given by an admin on my tests.
As a user, I want the program to save my test answers as I progress, but not commit the test answers until I choose to submit it
As an administrator, I want to be able to customize the header and footer of the page.
As an administrator, I want to be able to group questions into categories.
As an administrator, I want to be able to insert pictures into my questions.
As an administrator, I want to be able to import a test from a file and add it to the test bank
As a developer, I want to implement a test bank database
As an administrator, I want to run reports that are sent to the screen.
As a developer, I want to use Microsoft Azure Cloud for the database backend of my test system
As a developer, I want to create a user interface for the front end experience
As a developer, I want to have a code repository on github
As a developer, i want to create a production timeline

# Project Plan

## SCRUM Charts

The project was divided into the following sprints shown below. These sprints broke up the tasks by features that needed to be completed and assigned tasks to specific group members.

AT RISK	TASK NAME	FEATURE TYPE	RESPONSIBLE	STORY POINTS	START	FINISH	DURATION (DAYS)
<input checked="" type="checkbox"/>	Sprint 1				3/14/19	3/24/19	10
<input checked="" type="checkbox"/>	Feature 2	UI/UX Design	Ronnie, Drew, Sarim		3/14/19	3/24/19	10
<input checked="" type="checkbox"/>	Feature 3	Create UML plan of backend	Jordon, Kevin Z, Rahul		3/14/19	3/24/19	10
<input checked="" type="checkbox"/>	Feature 4	Create github repository	Jordon, Kevin Z, Rahul		3/14/19	3/24/19	10

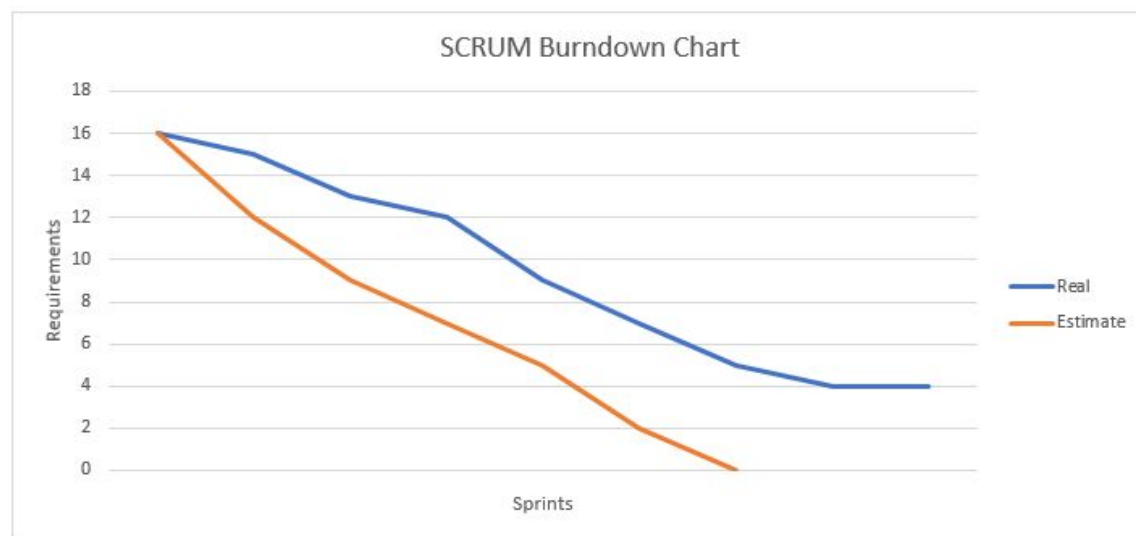
AT RISK	TASK NAME	FEATURE TYPE	RESPONSIBLE	STORY POINTS	START	FINISH	DURATION (DAYS)
<input checked="" type="checkbox"/>	Sprint 2				3/25/19	4/1/19	7
<input checked="" type="checkbox"/>	Feature 5	Make Java queries to manipulate database	Sarim, Kevin D		3/25/19	4/1/19	7
<input checked="" type="checkbox"/>	Feature 6	Begin implementation of UI	Ronnie, Drew, Sarim		3/25/19	4/1/19	7
<input checked="" type="checkbox"/>	Feature 7	Create classes	Jordon, Kevin Z, Rahul		3/25/19	4/1/19	7
<input checked="" type="checkbox"/>	Feature 8	Create login system	Jordon, Kevin Z, Rahul, Kevin D		3/25/19	4/1/19	7

### Sprint Chart 1.1

AT RISK	TASK NAME	FEATURE TYPE	RESPONSIBLE	STORY POINTS	START	FINISH	DURATION (DAYS)
<input type="checkbox"/>	Sprint 3				4/2/19	4/11/19	9
<input checked="" type="checkbox"/>	Feature 9	Create stored procedures, views, indexes	Sarim, Kevin D		4/2/19	4/11/19	9
<input checked="" type="checkbox"/>	Feature 10	Start to integrate UI with backend	Ronnie, Drew, Sarim		4/2/19	4/11/19	9
<input checked="" type="checkbox"/>	Feature 11	Start to code/flesh out classes	Jordon, Kevin Z, Rahul		4/2/19	4/11/19	9
<input checked="" type="checkbox"/>	Feature 12	Make different permissions for users/admins	Jordon, Kevin Z, Rahul, Kevin D		4/2/19	4/11/19	9

AT RISK	TASK NAME	FEATURE TYPE	RESPONSIBLE	STORY POINTS	START	FINISH	DURATION (DAYS)
<input type="checkbox"/>	Sprint 4				4/12/19	4/18/19	7
<input checked="" type="checkbox"/>	Feature 13	Implement MySQL server on Microsoft Azure	Sarim, Kevin D		4/12/19	4/18/19	7
<input type="checkbox"/>	Feature 14	Implement UI on Microsoft Azure	Ronnie, Drew, Sarim		4/12/19	4/18/19	7
<input checked="" type="checkbox"/>	Feature 15	Work on implementing connector classes in backend files for SQL	Jordon, Kevin Z, Rahul		4/12/19	4/18/19	7
<input type="checkbox"/>	Feature 16	Figure out how to host web app on Azure	Jordon, Kevin Z, Rahul, Kevin D		4/12/19	4/18/19	7

Sprint Chart 1.2



Scrum Chart 1.3



# Code Repository

All source code for this project can be found at this github link:

[https://github.com/rhl494/CB\\_V1](https://github.com/rhl494/CB_V1)

# Testing Plan and Results

We developed a variety of tests in order to ensure our app was implementing the functionality properly. They are listed along with their results below.

ID	Description	Input	Expected Result	Actual Result	Success/Fail
1	login as admin	correct username: "admin" correct password: "admin"	Brought to the home page	Brought to the home page	Success!
2	log in as user	correct username: "user" correct password: "user"	Brought to user home page	Brought to user home page	Success!
3	log in with incorrect credentials	username "h3110" password "no"	Error: "Access denied"	Error: "Access denied"	Success!
4	create new user from create user page	username "newguy" password "new" Account level "user"	User appears in "view users" with role "user"	User appears in "view users" with role "user"	Success!
5	create new admin from create user page	username "newadmin" password "topsecret" Account level "admin"	Admin appears in "view users" with role "admin"	Admin appears in "view users" with role "admin"	Success!
6	delete user	press delete button on user page	User no longer appears in "view users"	User no longer appears in "view users"	Success!
7	create a quiz	quiz name: "test" closing date: "5/3/2019" time limit (in minutes): "10" Passing Score: "10"	Creates quiz and takes you to the add question page	Creates quiz and takes you to the add question page	Success!
8	add question	question: "Who is Han Solo?" Answer: "Harrison Ford" Answer: "emperor" Answer: "vader" Answer: "jar jar"	Adds question to the quiz	Adds question to the quiz	Success!
9	View results of quizzes	click on results in the upper navigation bar	Shows results page	Shows results page	Success!

10	Change password	Old password: admin new password: admin2 confirm password: admin2	Password is changed	Password changes in database, but front end gives no message	Partial success
11	Edit question	Change any of the question fields	Question is updated	Question is updated	Success!
12	Delete question	Hit delete button on question in question list	Question is deleted	Question is deleted	Success!
13	Take a test as a user	Hit the take quiz button and select from the list of available quizzes	Quiz loads in with all questions	Quiz loads in with all questions	Success!
14	Submit quiz	Hit the submit button while taking a quiz	Quiz is finalized and the user is given their score	Quiz is finalized and the user is given their score	Success!

# Deployment Plan

## Overview

The website will be hosted on Microsoft's Azure platform; launching it will simply involve uploading the application to the host server. Since Azure is a cloud-based remote hosting service no server hardware will need to be installed to finish deployment.

## Scripts

SQL scripts will need to be used to create the tabular structure of the database and create the first administrative user to begin using the database.

## Deployment Plan

- A gradual cutoff will be used to move users to the new system. A minority of teachers will be moved to ChalkBoard so that the system can be parallelly tested with the old system.
- Professors teaching classes not reliant on technology (such as mathematics, english language) will be migrated first. Classes dependant on technology (computer science, engineering disciplines, etc.) will only be migrated until the system has been proven to work.
- Incremental deployment would not be ideal because users would need most of the planned features (checking grades, quickly writing and assigning tests, saving user results to the database) immediately to reduce users' stress and annoyance with the application.
- The system will be deployed during the summer to minimize the amount of users who could potentially lose labor.

### Rollback Plan

- In the event of a rollback, a minimal amount of data would need to be moved back to the old system since the application is gradually deployed to users who would be the least impacted by the program failing.
- By this design minimal labor will be lost if the system fails, if design flaws with the user interface prevent school staff from learning the new system, or if both occur and the system needs to be rolled back.
- Data added to ChalkBoard during its use can be exported as an CSV file. It can then be reformatted in a spreadsheet editor to be imported into the data of the old application.

# Maintenance Plan

- Deploy server to the cloud, which allows for having maintenance planned, scheduled, monitored and automated from a web portal.
- The cloud allows relevant information to be retrieved wherever it's located, alerting a technician to a problem before a breakdown.
- The cloud provider updates and maintains the server including the application software.
- The application manager must continue paying cloud fees to maintain and keep the server live, for both the online database and the application's front facing end.
- The application manager must occasionally delete and remove redundant data, old unused accounts, and expired quizzes.