



## CST-323 Design Report Template

<b>Topic:</b>	CLC - Milestone 3: Cloud Test Application on Cloud																																																			
<b>Date:</b>	12/1/2024																																																			
<b>Revision:</b>	3.0																																																			
<b>Team:</b>	1. Kaya Nelson																																																			
	2. Tyson Martin																																																			
	3.																																																			
	4.																																																			
	5.																																																			
	6.																																																			
<b>Weekly Team Status Summary:</b>	<table><thead><tr><th>User Story</th><th>Team Member</th><th>Hours Worked</th><th>Hours Remaining</th></tr></thead><tbody><tr><td><i>As a dev I would like to have Sprints, burndown chart, and repo setup for project management.</i></td><td>Kaya</td><td>1</td><td>0</td></tr><tr><td><i>As a user, I would like to have a way to Register and Login to the application</i></td><td>Kaya/Tyson</td><td>2</td><td>0</td></tr><tr><td><i>As a user, I would like to have a way to view all the app Posts</i></td><td>Kaya/Tyson</td><td>1</td><td></td></tr><tr><td><i>As a user, I would like a way to manage my account, update my password, and manage my posts</i></td><td>Tyson</td><td>2</td><td>2</td></tr><tr><td><i>As a stakeholder and dev, I would like to access this application from a cloud-hosted solution</i></td><td>Tyson</td><td>1</td><td>1</td></tr><tr><td><i>As a dev I would like a high level video overview of the code, webdesign, and features for presenting current progress.</i></td><td>Kaya</td><td>.5</td><td>0</td></tr><tr><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td></tr></tbody></table>				User Story	Team Member	Hours Worked	Hours Remaining	<i>As a dev I would like to have Sprints, burndown chart, and repo setup for project management.</i>	Kaya	1	0	<i>As a user, I would like to have a way to Register and Login to the application</i>	Kaya/Tyson	2	0	<i>As a user, I would like to have a way to view all the app Posts</i>	Kaya/Tyson	1		<i>As a user, I would like a way to manage my account, update my password, and manage my posts</i>	Tyson	2	2	<i>As a stakeholder and dev, I would like to access this application from a cloud-hosted solution</i>	Tyson	1	1	<i>As a dev I would like a high level video overview of the code, webdesign, and features for presenting current progress.</i>	Kaya	.5	0																				
User Story	Team Member	Hours Worked	Hours Remaining																																																	
<i>As a dev I would like to have Sprints, burndown chart, and repo setup for project management.</i>	Kaya	1	0																																																	
<i>As a user, I would like to have a way to Register and Login to the application</i>	Kaya/Tyson	2	0																																																	
<i>As a user, I would like to have a way to view all the app Posts</i>	Kaya/Tyson	1																																																		
<i>As a user, I would like a way to manage my account, update my password, and manage my posts</i>	Tyson	2	2																																																	
<i>As a stakeholder and dev, I would like to access this application from a cloud-hosted solution</i>	Tyson	1	1																																																	
<i>As a dev I would like a high level video overview of the code, webdesign, and features for presenting current progress.</i>	Kaya	.5	0																																																	



GRAND CANYON  
UNIVERSITY™

<b>GIT URL:</b>	<a href="https://github.com/Kdeshun/CST323-Milestone">https://github.com/Kdeshun/CST323-Milestone</a>			
<b>Peer Review:</b>	<a href="#">Y/N</a>	We acknowledge that our team has reviewed this report and we agree to the approach we are all taking.		

Deployment URL: <http://cst323clc-env.eba-mm2xi3h5.us-east-1.elasticbeanstalk.com/>

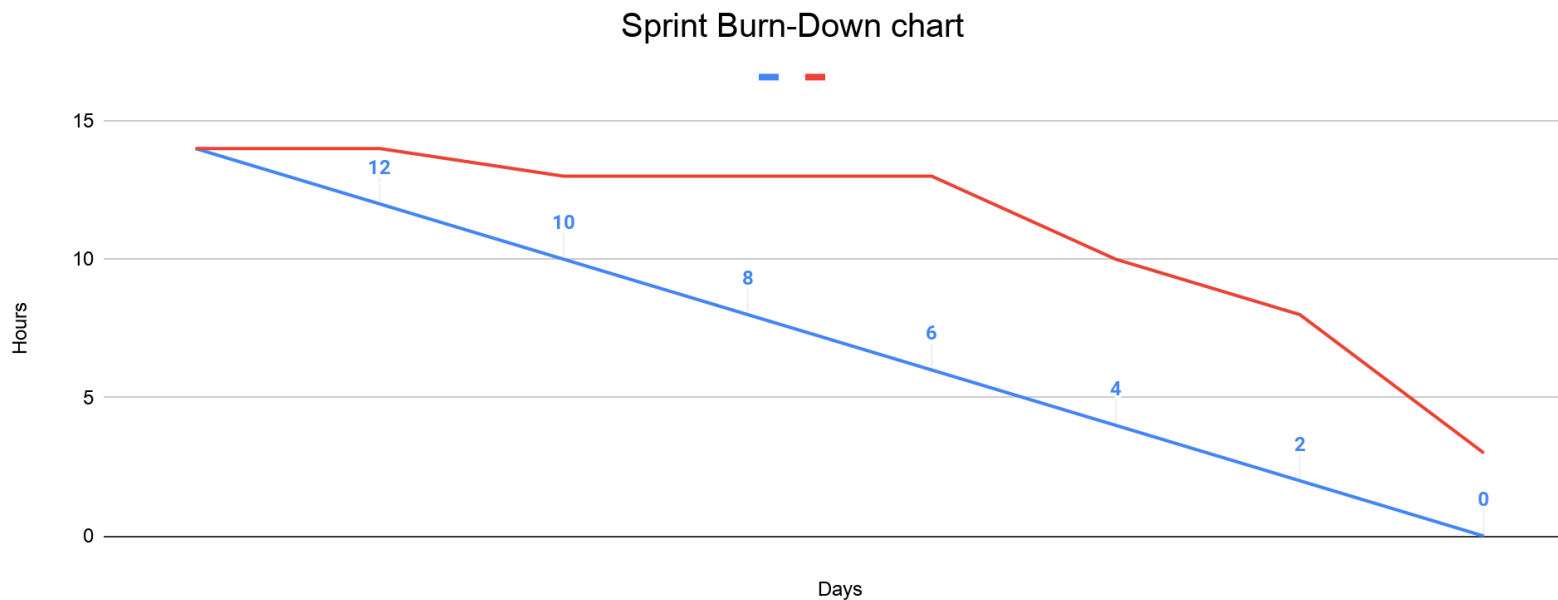
Presentation URL: <https://go.screenpal.com/watch/cZlhhlNnlkq>



## Planning Documentation

### Agile Scrum Burn Down Chart:

*This needs to contain a URL to Bitbucket Scrum Burn Down Chart Artifact.*



### Agile Retrospective Results:

*The following table should be completed after each Retrospective on Things That Went Well (Keep Doing). An alternative to the following table is to use a Mind Mapping tool such as Coggle. If you use a Mind Mapping tool, you must include a URL or Image File.*

What Went Well
Having a centralized method of communication helped communicate team standards to the group on how to complete and submit work.



The initial setup with react/ node/ npm/ and base project file worked flawlessly

*The following table should be completed after each Retrospective on Things That Didn't Go Well (Stop Doing) and What Would Be Done Differently Next Time with an Action Plan to Improve (Try Doing and Continuous Improvement). An alternative to the following table is to use a Mind Mapping tool such as Coggle. If you use a Mind Mapping tool, you must include a URL or Image File.*

What Did Not Go Well	Action Plan	Due Date

## Design Documentation

### Install Instructions:

*Include step-by-step instructions for setting up your database, configuring, and deploying/installing your application. This section should also include detailed instructions for what configuration files are required by your application, what configuration settings need to be adjusted for various runtime (development or production) environments, and where the files need to be deployed to. This section should also contain detailed instructions for how to clone your application source code from BitBucket and deploy the application to an externally hosted site.*

The initial step involves cloning the repository to your local machine. Once the cloning is complete, navigate to the /appdirectory. Open a terminal window and execute npm install to install the necessary dependencies. Next, switch to the /client directory and run npm install (or simply npm i) once more.

At this stage, all required dependencies should be properly installed and linked. You can launch the development server by returning to the /app directory and executing npm run start. This command will initiate the development server defined in server.js and start the React script process located in the /client directory.

To prepare the application for deployment, navigate to the /client directory again and run npm run build. Wait for the build process to complete. If you deploy the app as a Node.js application on Heroku or a similar platform, that service's build system will handle dependency installation at the root /app level and start the application server using npm run start.

### General Technical Approach:

*In your own words describe your approach and design here. You should also summarize any meeting notes, brain storming sessions, and so forth that you want to retain thru the design of your project.*

Our team is focused on developing a blogging platform that is accessible to the open web. To facilitate collaboration, we will utilize a range of tools. Discord will serve as our primary communication channel, allowing us to stay connected and work together on projects throughout the week. For source code management and version control, we will rely on GitHub. Google Docs will be used for maintaining sprint and burndown charts, as well as for updating our design reports. For wireframing and UML designs, we will employ Draw.io. Additionally, we will create ER diagrams using MySQL Workbench to ensure our database design is well-structured.

### Key Technical Design Decisions:

*Any final technical design decisions, (e.g., framework decisions) should be documented here. List the technology/framework, its purpose in the design, and why it was chosen.*

### Non-Code/Team Tools



## GRAND CANYON UNIVERSITY™

- Discord: For team communication.
- Google Docs: For documentation.
- GitHub: For version control.
- Visual Studio Code: Preferred code editor, though alternatives can be used.
- Loom: For video presentations.

### Framework/Language

The application will utilize a pure JavaScript stack:

- React: Frontend development framework.
- NodeJS: Backend server-side development framework.
- MySQL: Web database engine, used by companies like Uber.

### Libraries/Packages

To enhance the functionality of the React/NodeJS stack, the following libraries and packages will be utilized:

- MySQL: [NPM Package](#)
- EditorJS: [Official Site](#)
- Axios: [NPM Package](#)
- UUID: [NPM Package](#)
- React Bootstrap: [Official Documentation](#)

This setup provides a robust foundation for developing a basic blogging platform with effective communication and project management tools.

### Known Issues:

*Any anomalies or known issues in the code or functionality should be documented here.*

Currently, when new users register, they should be redirected to the login page for immediate access. This is a minor issue that will be addressed in the next milestone. Additionally, another known issue is that the “write/edit posts” button on the Account page needs to be styled to better match the aesthetics of the change password form.

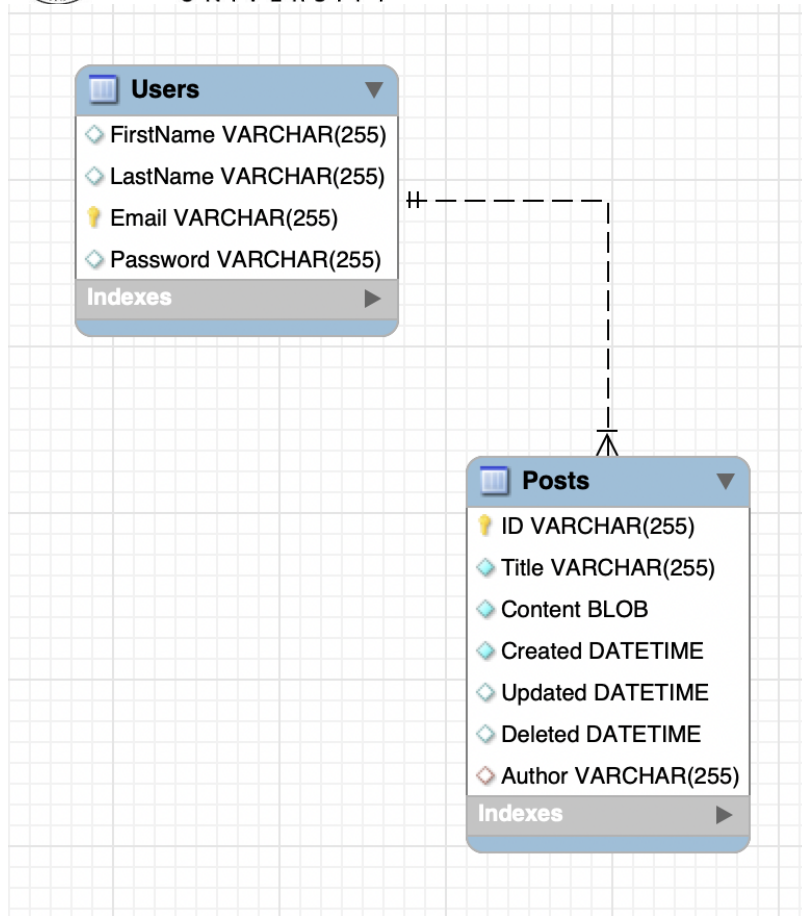
### Risks:

*Any risks, unknowns, or general project elements that should be tracked for risk management should be documented here.*

No known Risks at this time.

### ER Diagram:

Below is an image of the current database ER diagram.

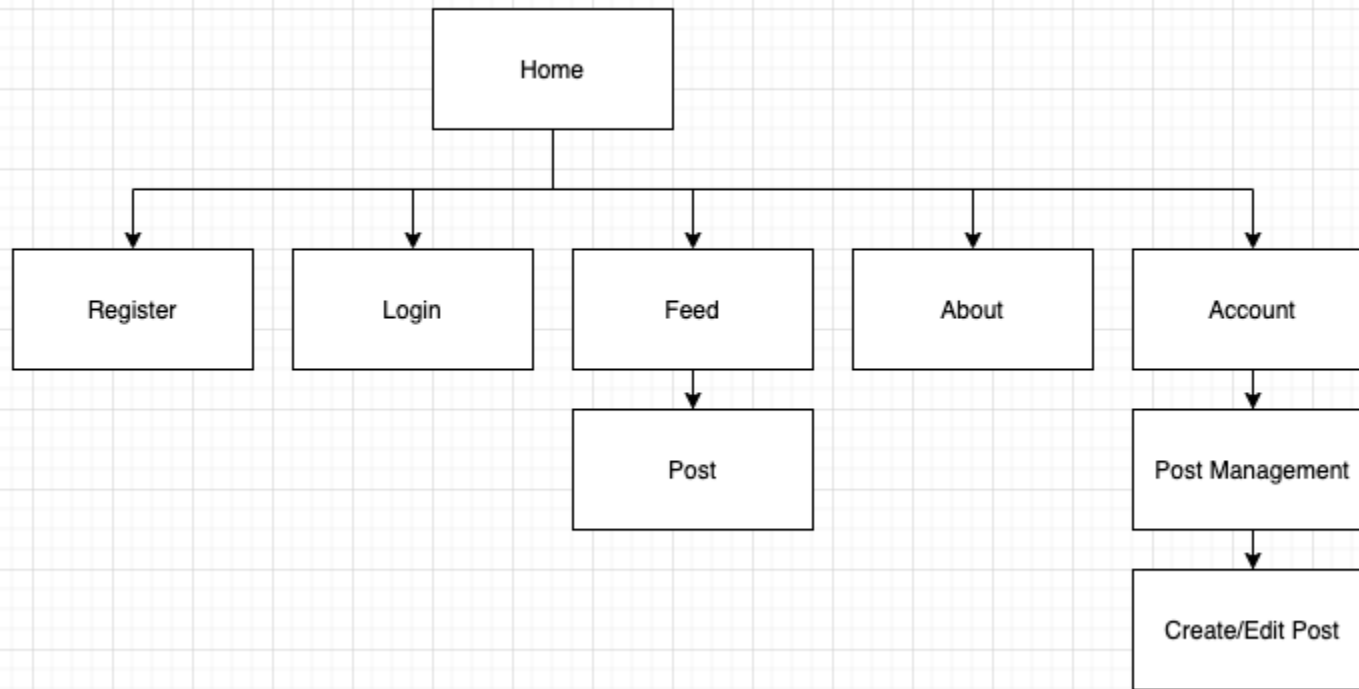


### DDL Scripts:

The current DDL Create script can be found at the following link <https://github.com/Kdeshun/CST323-Milestone>

### Sitemap Diagram:

*Include an image file of your Sitemap diagram.*







GRAND CANYON  
UNIVERSITY™

**User Interface Diagrams:**

*You should insert any wireframe drawings or white board concepts that were developed to support your application. If you have no supporting documentation, please explain the rationale for labeling this section N/A.*



BACK

## EDIT POST

Title

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

DELETE POST

SAVE CHANGES



BACK

## CREATE ACCOUNT

FIRST NAME

John

EMAIL

Smith

EMAIL

eg. name@gmail.com

PASSWORD

\*\*\*\*\*

REGISTER



BACK

## TITLE OF POST

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

ALL POSTS



BACK

## POST MANAGEMENT

WRITE NEW POST

EDIT A POST

POST TITLE

POST TITLE

POST TITLE

POST TITLE



BACK

## MY ACCOUNT

WRITE OR EDIT POSTS

FIRST NAME

John

LAST NAME

Smith

EMAIL

eg. name@gmail.com

PASSWORD

\*\*\*\*\*

SAVE CHANGES



BACK

## LOGIN

EMAIL

eg. name@gmail.com

PASSWORD

\*\*\*\*\*

LOG IN

Don't have an account?

REGISTER



GRAND CANYON  
UNIVERSITY™

THE "BLOGS ARE COOL" BLOG

[HOME](#)

[ABOUT](#)

[FEED](#)

[ACCOUNT](#)

[LOGIN](#)

# Read interesting things, even if you don't want to.

[CREATE ACCOUNT](#)

[LOGIN](#)





BACK

## ALL ARTICLES

TITLE OF ARTICLE

TITLE OF ARTICLE

TITLE OF ARTICLE

TITLE OF ARTICLE

### Class Diagrams:

*You should insert any class diagrams here. Your class diagrams should be drawn correctly with the three appropriate class compartments, + and – minus to indicate accessibility, and the data types for the state/properties as well as method arguments and return types. If you have no supporting documentation, please explain the rationale for labeling this section N/A.*

## UML Diagram:

No UML diagram at this time, while we sort out how we will organize and structure our code layers between the front/back ends.

## Service API Design:

*This section should fully document any Third Party Service Interface API's being consumed or application specific Service API's being published, how to access the service, what parameters are required by the API, and the detailed JSON data format specification that could be used by a third party developer to integrate with the service and API.*

At this milestone, we do not anticipate needing to use any third-party APIs or services.

### Service API

An API layer will be established between the web client and the database, accessible at the route /api on port 3001. This layer will provide REST routes for interacting with both User and Post records.

#### Users

- **GET** /user/:email
- **GET** /users
- **POST** /user/create
- **POST** /user/update
- **POST** /user/auth

#### Posts

- **GET** /post/:id
- **GET** /posts
- **GET** /posts/:user
- **POST** /post/create
- **POST** /post/update

Additional documentation detailing the expected parameters for POST requests and the return objects for each route will be added to this documentation soon.

## Security Design:

*This section should outline the design for how authentication and authorization was supported. This section should also contain all of the roles and privileges that are supported by the design.*



Users will log into their accounts using their email and a unique password. We will salt and hash these passwords before storing them. During a login attempt, the inputted password will be salted and hashed, then compared to the stored hash. This approach ensures that raw passwords are never stored in the database.

Users will also have the ability to update their passwords, provided they enter their current password.

To handle password reset requests, we will implement the following process:

**1. Request Submission:**

- Users will submit a password reset request by providing their registered email address.

**2. Email Verification:**

- Upon receiving the request, the system will generate a secure, unique token and send a password reset email to the user's registered email address. This email will include a link to reset their password, which contains the token.

**3. Token Validation:**

- When the user clicks the link, they will be directed to a password reset page. The system will verify the token to ensure it is valid and has not expired.

**4. Password Reset:**

- After token validation, the user will be prompted to enter a new password. This new password will be salted and hashed before being stored in the database.

**5. Confirmation:**

- Once the password has been successfully updated, the user will receive a confirmation message, and the token will be invalidated to prevent reuse.

This process enhances security by ensuring that password resets are handled safely and efficiently.

**Other Documentation:**

*You should insert any additional drawings, storyboards, white board pictures, project schedules, tasks lists, and so forth that support your approach, design, and project. If you have no supporting documentation, please explain the rationale for labeling this section N/A.*