**CST-341 Design Report Template**

|  |  |  |
| --- | --- | --- |
| **Topic:** | Topic 4: Introduction to Spring Framework Part 3 - Spring Java Database Connectivity (JDBC) and Spring Security | |
| **Date:** | 09/13/2020 | |
| **Revision:** | 1.0 | |
| **Team:** | 1. Adam Brown | |
| 1. Tim James | |
| 1. Sam Overson | |
| 1. Calista Cartwright | |
|  | 1. Nicholas Peron | |
|  | 1. Casey Huz | |
| **Weekly Team Status Summary:** | |  |  |  |  | | --- | --- | --- | --- | | **User Story** | **Team**  **Member** | **Hours**  **Worked** | **Hours Remaining** | | *As a dev I would like to have Sprints, burndown chart, and repo setup for project management.* | *Tim* | *1* | *0* | | *As a dev I need to have a good register page design* | *Tim* | *1* | *0* | | *As a dev I need to have a good home page design* | *Tim* | *1* | *0* | | *As a dev I need to have a good login page design* | *Sam* | *1* | *0* | | *As a dev I need to have a good my profile page design* | *Casey* | *1* | *0* | | *As a dev I need to have peer review of my code.* | *Everyone* | *2* | *0* | | *As a dev I would like to use Spring JDBC* | *Nick, Adam, Calista* | *3* | *0* | |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |  | | |
| **GIT URL:** | [*https://github.com/tjames222/CST-341CLC*](https://github.com/tjames222/CST-341CLC) | |
| **Peer Review:** | *Y/N* | We acknowledge that our team has reviewed this report and we agree to the approach we are all taking. |

**Planning Documentation**

**Agile Scrum Product Backlog:**

*This needs to contain a URL to Bitbucket Scrum Product Backlog Artifact.*

[*https://github.com/tjames222/CST-341CLC/projects*](https://github.com/tjames222/CST-341CLC/projects)

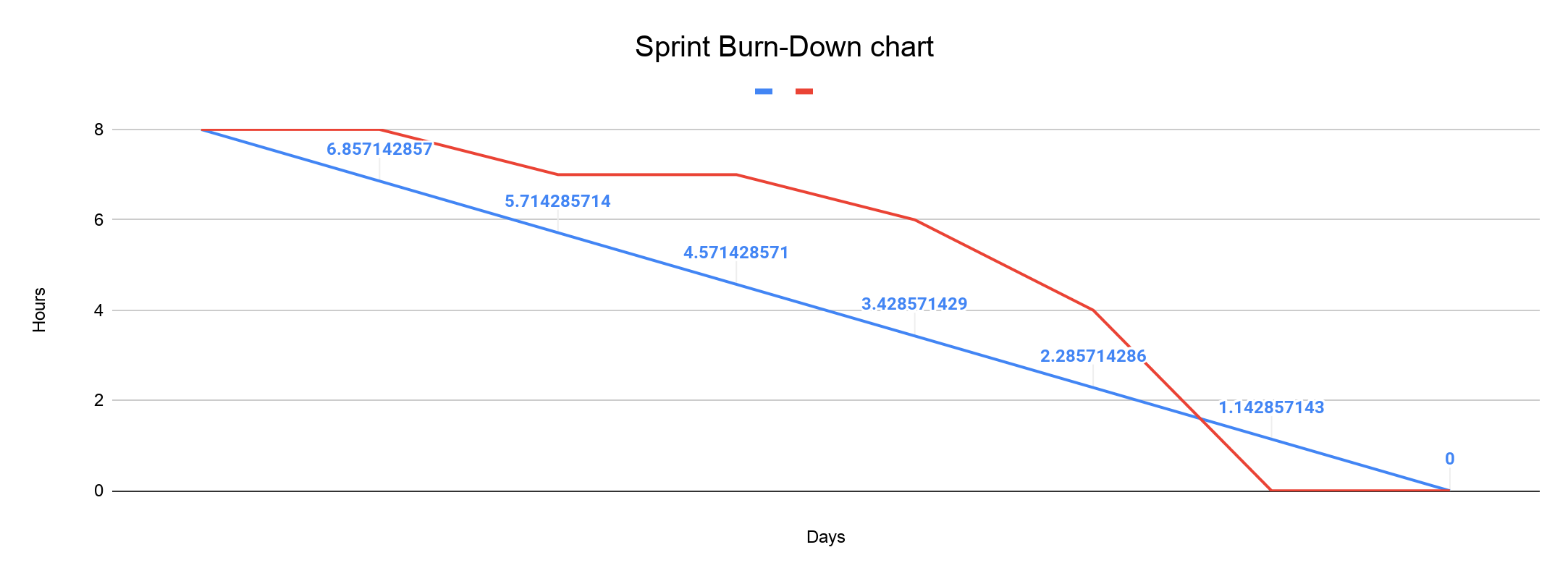
**Agile Scrum Sprint Backlog:**

*This needs to contain a URL to Bitbucket Scrum Sprint Backlog Artifact. This current week’s progress should be reflected in the above section of this Design Report.*

[*https://github.com/tjames222/CST-341CLC/projects/1*](https://github.com/tjames222/CST-341CLC/projects/1)

**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** |
| The team continued to communicate well and assist each other on difficult parts of the assignments. |
| The individual assignment was much easier to complete this week, allowing for more time to focus on the group project. |
|  |

*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** |
| The team had issues ensuring the proper version of the project was being pulled by team members prior to working on their assigned tasks. | Ensure there is better communication between members before and after submissions to the repo are made. | 9-20-20 |
|  |  |  |
|  |  |  |

**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.*

**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 approach is to design a blog website. We will work together using a variety of tools to help collaboration efforts. We will use Discord to stay in touch throughout the week and collaborate on projects. We will use GitHub for source code management and version control. We will use Google docs for sprint and burndown charts as well as Design report updates.

**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.*

Our team will design a blog website.

Discord will be used for communication.

Google docs will be used for documentation.

GitHub will be used for version control.

Eclipse IDE will be used for source code.

Loom will be used for video presenting.

**Known Issues:**

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

Some users are not able to get the database to function correctly, which does not allow for User or Post objects to be accessed.

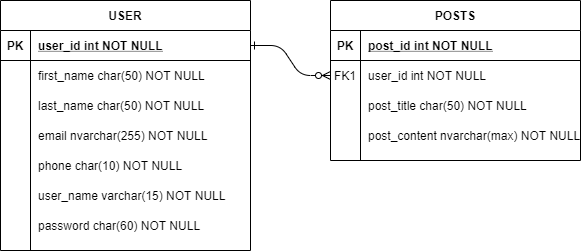
**Risks:**

*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:**

*Include an image file of your ER database diagram.*

**

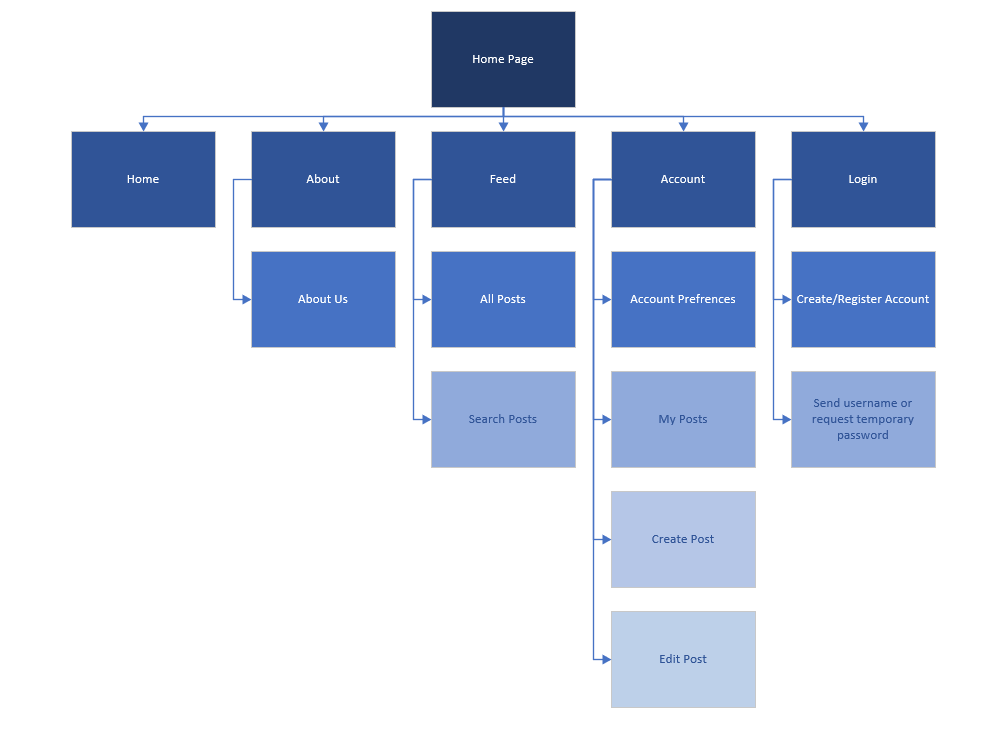
**DDL Scripts:**

*This should contain a link to Bitbucket from where the DDL script can be downloaded.*

<https://github.com/tjames222/CST-341CLC/blob/dev/blog/src/model/DDL.xsd>

**Sitemap Diagram:**

*Include an image file of your Sitemap diagram.*



**User Interface Diagrams:**

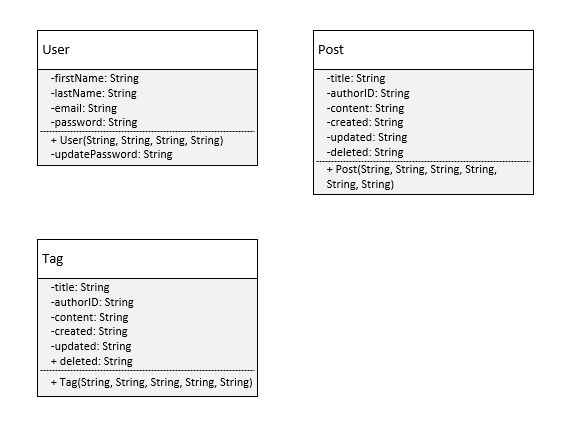
*You should insert any wireframe drawings or white board concepts that were developed to support your application. If you have no supporting documentation, please explain the rationale for labeling this section N/A.*

**

**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:

**

**Service API Design:**

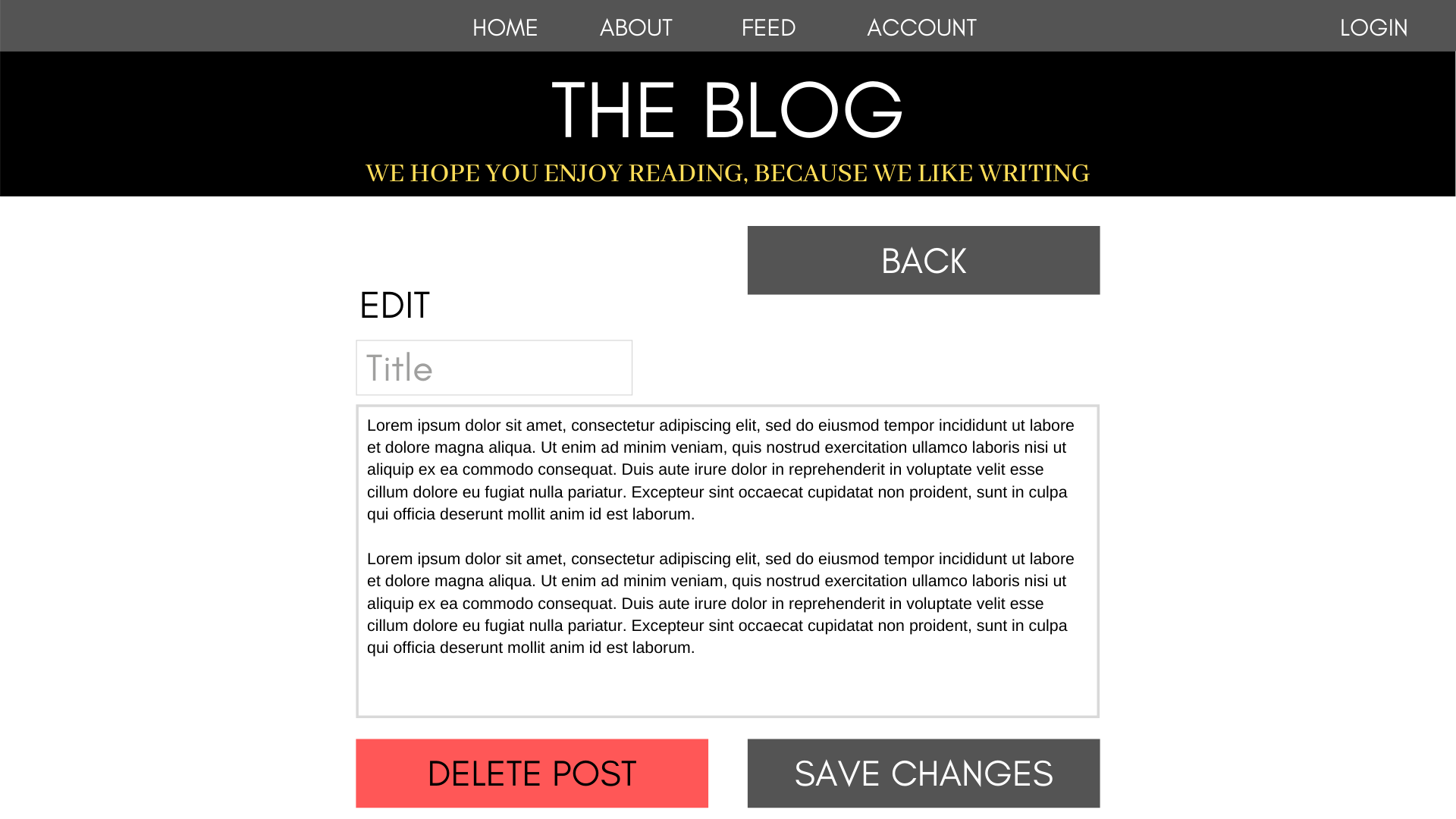
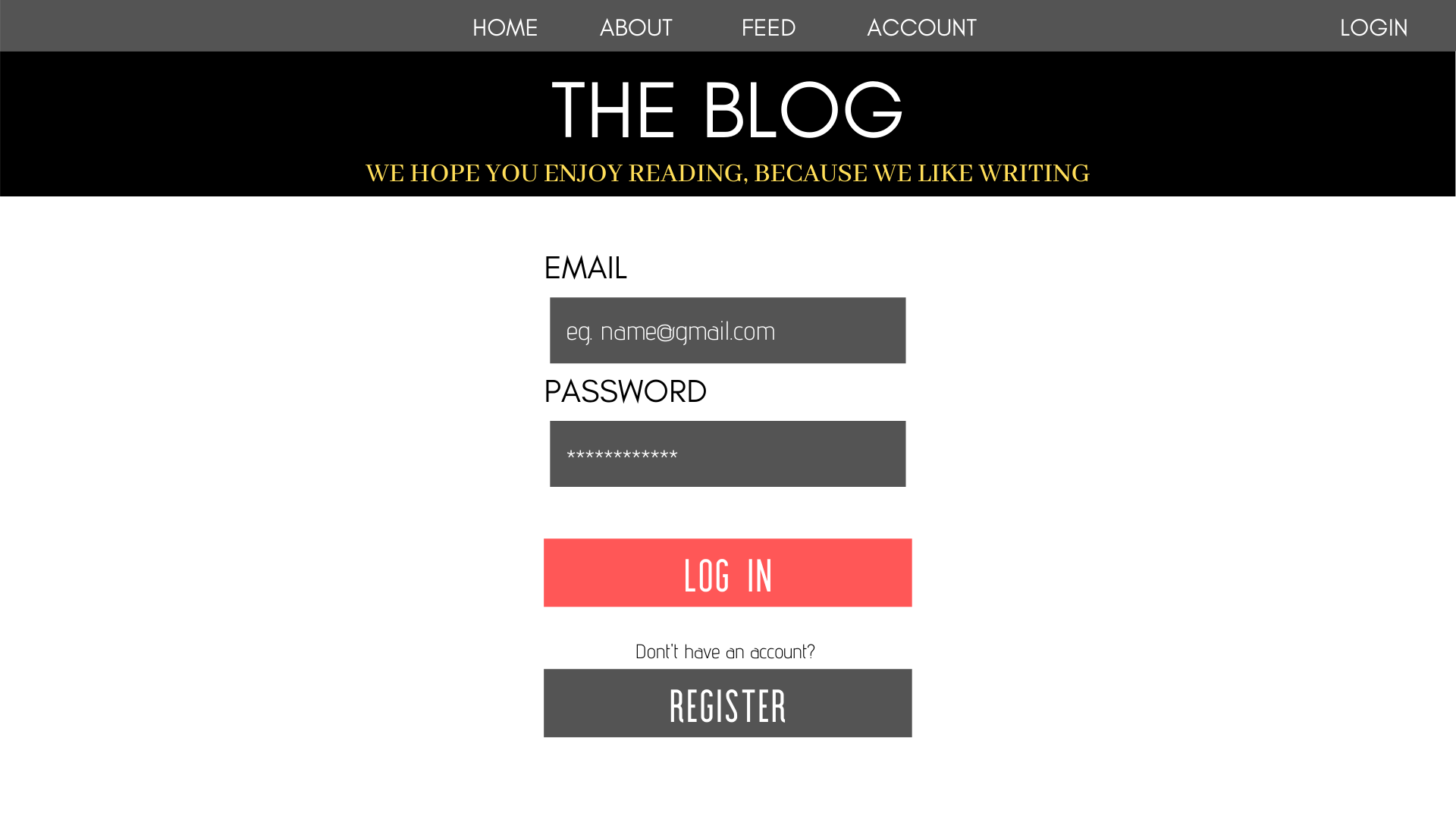
*This section should fully document any Third Party Service Interface API’s being consumed or application specific Service API’s being published, how to access the service, what parameters are required by the API, and the detailed JSON data format specification that could be used by athird party developer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

**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.*

**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 laveling this section N/A.*

**