**CST-361 - Design Report Template**

|  |  |  |
| --- | --- | --- |
| **Topic:** | *CLC Milestone 4* | |
| **Date:** | *02/10/2019* | |
| **Revision:** | *1.7* | |
| **Team:** | 1. *Michael Weaver* | |
| 1. Kurt Newcomb | |
|  | |
|  | |
| **Weekly Team Status Summary:** | |  |  |  |  | | --- | --- | --- | --- | | **User Story** | **Team**  **Member** | **Hours**  **Worked** | **Hours Remaining** | | *Reworked Restful service* | *Michael* | *2* |  | | *Business service implementation* | *Michael* | *2* |  | | *Data service refactoring* | *Kurt* | *4* |  | | *General Refactoring* | *Team* | *4* |  | |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |  | | |
| **GIT URL:** | *https://github.com/battousairurik/CST361\_Design\_Patterns\_In\_Java* | |
| **Peer Review:** | *Y* | We acknowledge that our team has reviewed this report and we agree to the approach we are all taking. |

**Planning Documentation**

**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** |
| **After teaming up with Kurt the development process went much smoother** |
| **The data access classes were specialized for the letters and users** |
|  |

*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** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**Note: Code will be sent to you via email and will be available on the github link**

**Design Documentation**

**General Technical Approach:**

*You should, in your own words, describe your approach and design here. You should also summarize any meeting notes, brainstorming sessions, etc. that you want to retain through the design of your project.*

The application is set up as a dynamic web project. It is broken up with the MVC architecture. The views are handled by the forms controller, which routes the request based upon data entered. The forms controller utilizes either the letter service or the user service based upon the user input. Restful services then catch either the user EJB or the letter EJB. Additionally, these are intercepted anytime a method is called by our custom interceptor. A singleton logging service is then called to document the application activity. The EJB then interact with DAO classes which handle database access and then pass back the data requested.

**Key Technical Design Decisions:**

*Any final technical design decisions, such as framework decisions etc., should be documented here. This should list the technology/framework, its purpose in the design, and why it was chosen.*

MVC architecture, Singleton design pattern for Logger, Dependency injection, intercetors.

**Known Issues:**

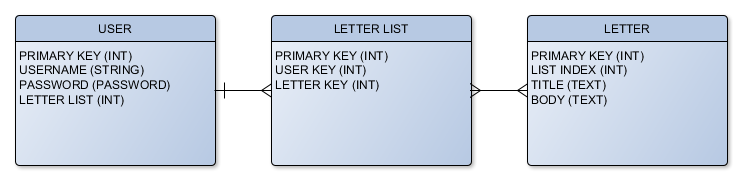
*Any anomalies or known issues in the code or functionality should be documented here.*

**Risks:**

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

**ER Diagram:**

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

**

**DDL Scripts:**

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

**Flow Charts:**

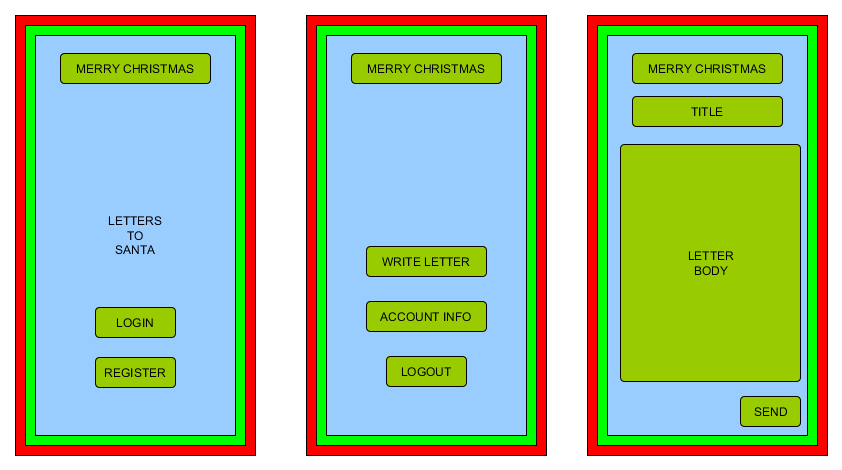
*You should insert any flowcharts here. Flowcharts should document algorithms or workflow that will be implemented in your program. At a minimum, this should contain a flowchart of the Minesweeper game logic.*

**Sitemap Diagram:**

*Image file of your Sitemap diagram.*

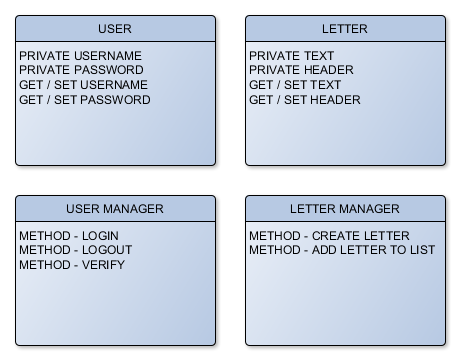
**User Interface Diagrams:**

*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 leaving this section as 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 leaving this section as N/A.*



**Security Design:**

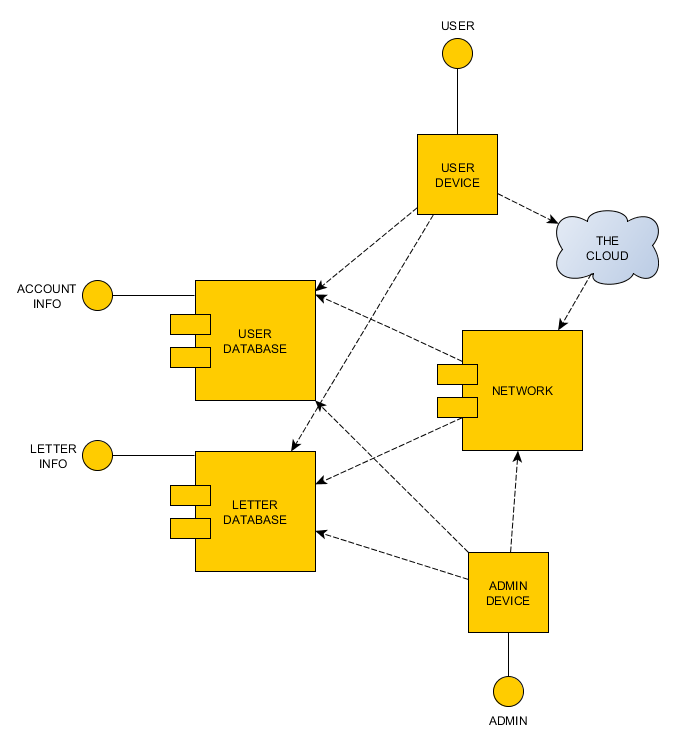
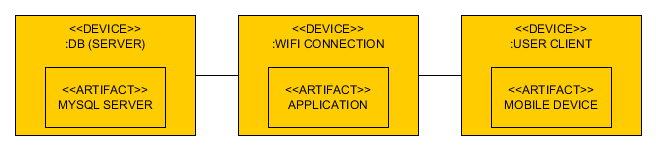
*This section should outline the design for how authentication and authorization was supported. This section should also contain all of the roles and privileges that are supported by the design.*

**Pseudo Code:**

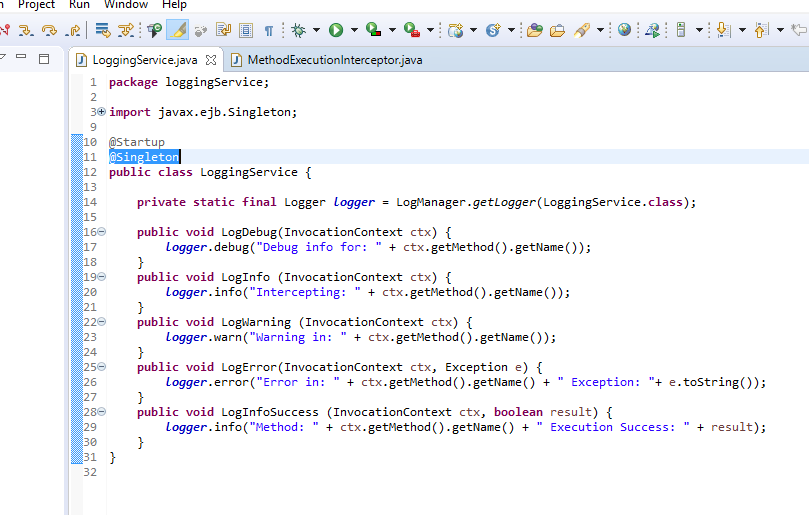
*You should provide Bitbucket URL references to any code stubs and pseudo code. If you have no supporting documentation, please explain the rationale for leaving this section as N/A.*

**Other Documentation:**

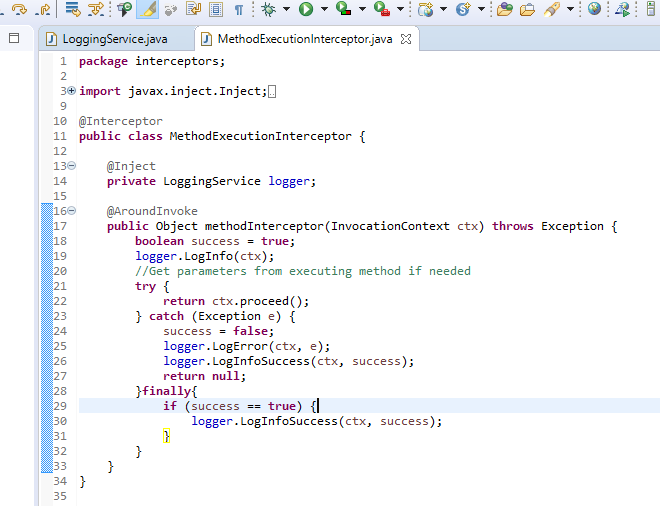
*You should insert any additional drawings, storyboards, white board pictures, project schedules, tasks lists, etc. that support your approach, design, and project. If you have no supporting documentation, please explain the rationale for leaving this section as N/A.*

******

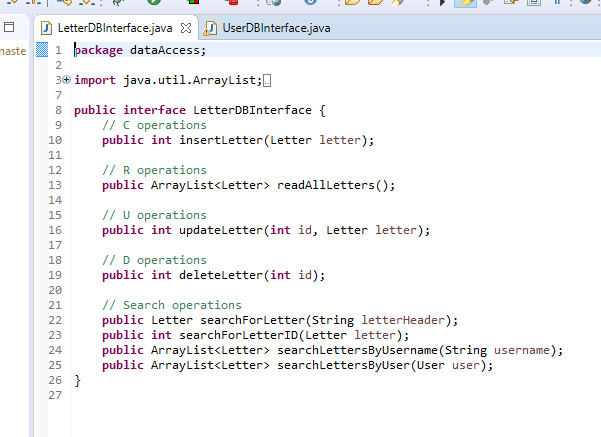
SINGLETON LOGGING SERVICE



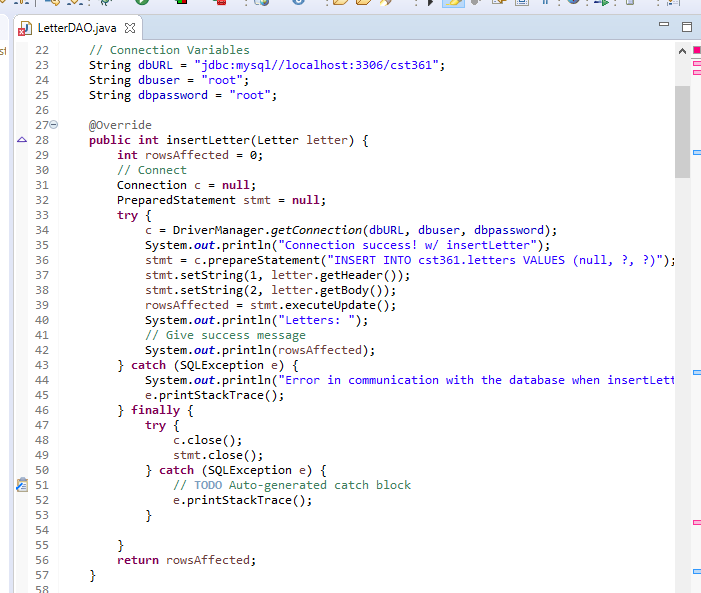
EJB METHOD INTERCEPTOR



LETTER DAO



LETTER DAO EXAMPLE



USER DAO

