

SlangDictionary

Project report

Student:

Bleard Rexhaj, ID: 611019

Erdenesaikhan Tserendavga, ID: 611104

Enkhjargal Gansukh, ID: 611031

Team name: Captains

Team slogan: None of us is as smart as all of us

Professor:

Joe Bruen

Course:

Enterprise Architecture CS544, August block of 2020

CAPTAINS - SlangDictionary

Revision History

Date	Issue	Description	Author
<08/10/20>	<0.1>	Create new document	Erdenesaikhan Tserendavga
<08/14/20>	<0.2>	User Module written	Bleard Rexhaj
<08/14/20>	<0.3>	Term and Rest Module	Enkhjargal Gansukh
<08/14/20	<0.4>	Add Category Module	Erdenesaikhan Tserendavga

CAPTAINS - SlangDictionary

Table of Contents

1.	Introduction	4
1.1	Purpose.....	4
2.	Requirements - Use-Case – Usage Scenarios	5
2.1	Register User.....	5
2.2	Add Term	6
2.3	Search Term by Word	6
3.	High Level Design	7
3.1	User Module.....	7
3.2	Term Module.....	8
3.3	Category Module	8
4.	Detailed Design	9
4.1	User Usage	10
4.2	User Sequence Diagram.....	11
4.3	Term List Sequence Diagram	12
4.4	Add Term Sequence Diagram	13
4.5	Category Sequence Diagram	13
5.	Design Alternatives.....	14
6.	Issues, Risk and Dependencies	14
7.	Future Considerations	14
8.	References	14

CAPTAINS - SlangDictionary

1. Introduction

Slang dictionary is a project where users can simply browse new slang words every day. The web application “SLANG DICTIONARY” will provide an easy and convenient way for users to register and add slang words, the normal guest users can simply browse through and read about the words they are interested in. Registration will require a valid Email address for identity verification, guests will be able to add comments and to vote regarding the words, term adders will be able to edit their slang added words.

1.1 Purpose

To learn and implement some of the Main Technologies that have been covered in the course lectures and to broaden our knowledge enhancing the capability of interpreting theory into practice. To be able to understand and design better software applications using combination of different tools and technologies.

1.2 Scope

Slang Dictionary project introduces a service layer exposed through the Spring MVC Controller and Restful Controller as well. It has one Spring Email Service that will send emails to specific users. There exists the option to get slang words from the web app through the Rest Controller.

1.3 Technologies

To achieve the mentioned functionality, we applied technologies such as

- Spring boot
- Hibernate JPA
- Email
- Spring Security
- AOP

CAPTAINS - SlangDictionary

- Testing
- MySQL
- GitHub
- Hibernate Performance
- Validation

2. Requirements - Use-Case – Usage Scenarios

2.1 Register User

Intent	Registering a user to the system
Description	The user will be validated, a password encryption and roles will be added to him through AOP-UserAspect.java, and after succession an email will be sent to the user using AP-UserAspect.java.
Preconditions	User should not exist, and data fields should be filled in.
Success Post Condition	User Created Successfully.
Failed Post Condition	Error message will be displayed.
MAIN FLOW	
Step	Action
1	User Data is received and checked for validation errors.
2	User is assigned roles and password encryption with AOP.
3	An email is sent to the user for account validation.
4	If there is a failure and error message will be displayed.

CAPTAINS - SlangDictionary

2.2 Add Term

Intent	Member add term to dictionary
Description	The user needs to be logged in and insert all inputs in Add Term Page. Inputs will be validated and insert into database.
Preconditions	User should be logged in, and add term data fields should be filled in.
Success Post Condition	Term added in database and redirect to Home page.
Failed Post Condition	Error message will be displayed.
MAIN FLOW	
Step	Action
1	Check user logged in and select Add Term menu.
2	User fill in input fields and validate all inputs.
3	Save the term in database.
4	Redirect to Home Page.

2.3 Search Term by Word

Intent	Find terms by word
Description	The user can search terms by word they want. The application can find by inserted word from the database.
Preconditions	User should insert word in search input field.
Success Post Condition	Found terms will be shown by listing.
Failed Post Condition	Error message will be displayed.
MAIN FLOW	

CAPTAINS - SlangDictionary

Step	Action
1	User inserts word in search input field and click search button.
2	System find word in database return term list.
3	Display found list of terms.

3. High Level Design

3.1 User Module

Description of technologies and their roles for the User Module:

- Validation – Validation was made by spring validation, main validation annotation used are @NotEmpty, @Email and custom validations like @EmptyOrSize, @FieldMatch that will check whether the password and confirm password will match.
- Spring Email - Spring Email was used in order to complete the implementation of the user, so whenever a user is registered it will be in a disabled state, an email will be generated and sent out and when the user clicks the generated link that includes the token his data will be enabled and ready for use.
- AOP – Aspect oriented programing was used in the User Module with the @Before and @After annotations, there was the need to encrypt the password and add a default role before the user was being persisted to the database and after persistence, to send an confirmation email to the user in order for him to confirm his account, and aop was the perfect tool to solve this cross-cutting concern.
- Security - Spring security was user to authorize the user based on his authorities.
- ORM/Hibernate - Hibernate was used to persist ORM java objects.

CAPTAINS - SlangDictionary

- Testing – `UserIntegrationTest` was created to test whether a user will be registered successfully, and whether a user will be able to access a protected path(`MockMvcUser`).

3.2 Term Module

Term module is used in Home, My Term list and Rest API pages. This module's main purposed term add, list and search logics to show users and guests by JSP pages.

3.3 Category Module

Category module provide list of categories in category page. The basic requirement for a category page is to categorize how many categories users have. A list of words containing the user-selected category is loaded on the category list page. The developer can add the required categories to the system, and the user cannot add new ones directly.

CAPTAINS - SlangDictionary

4. Detailed Design

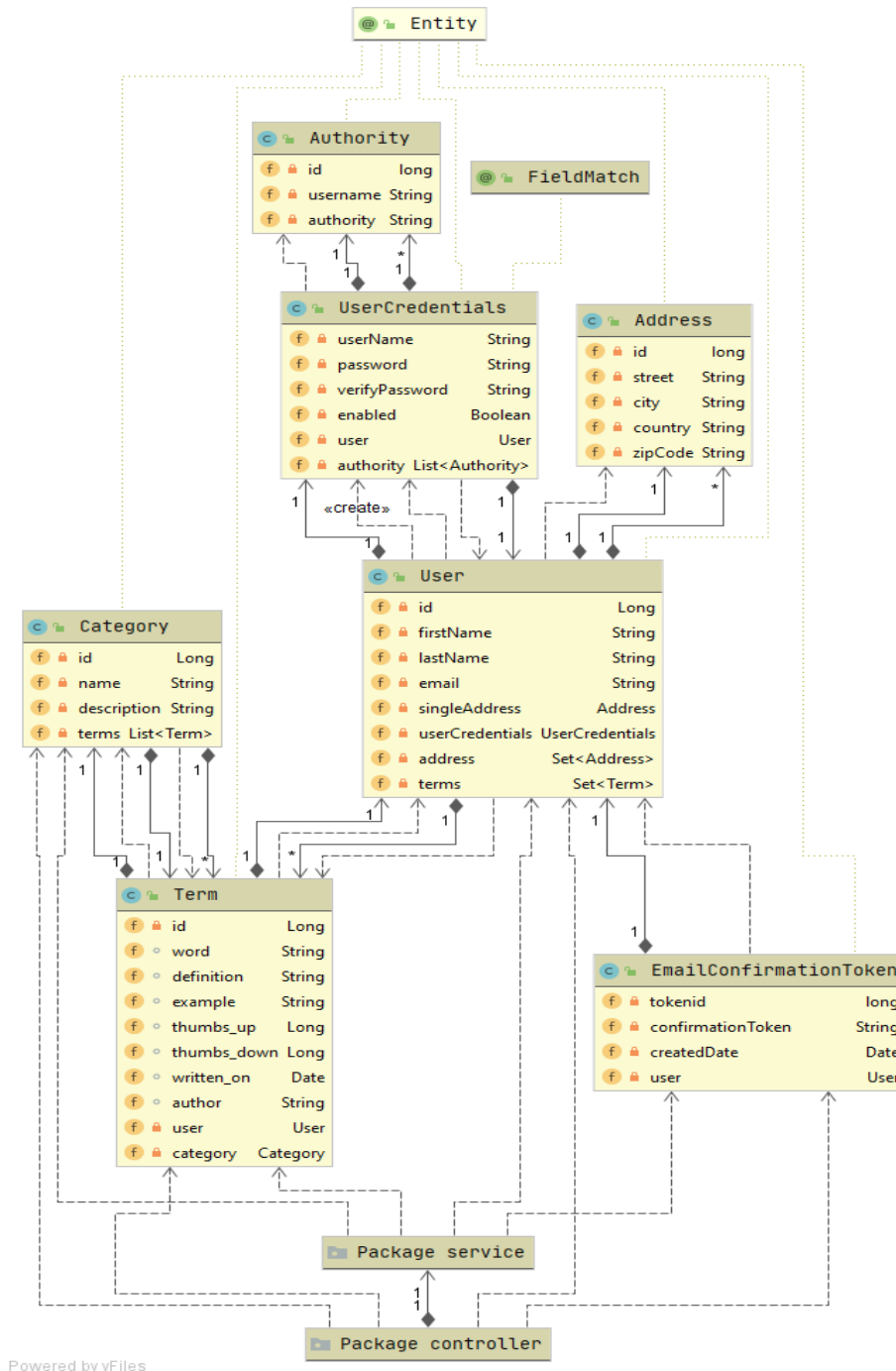


Figure 1. Class diagram

CAPTAINS - SlangDictionary

4.1 User Usage

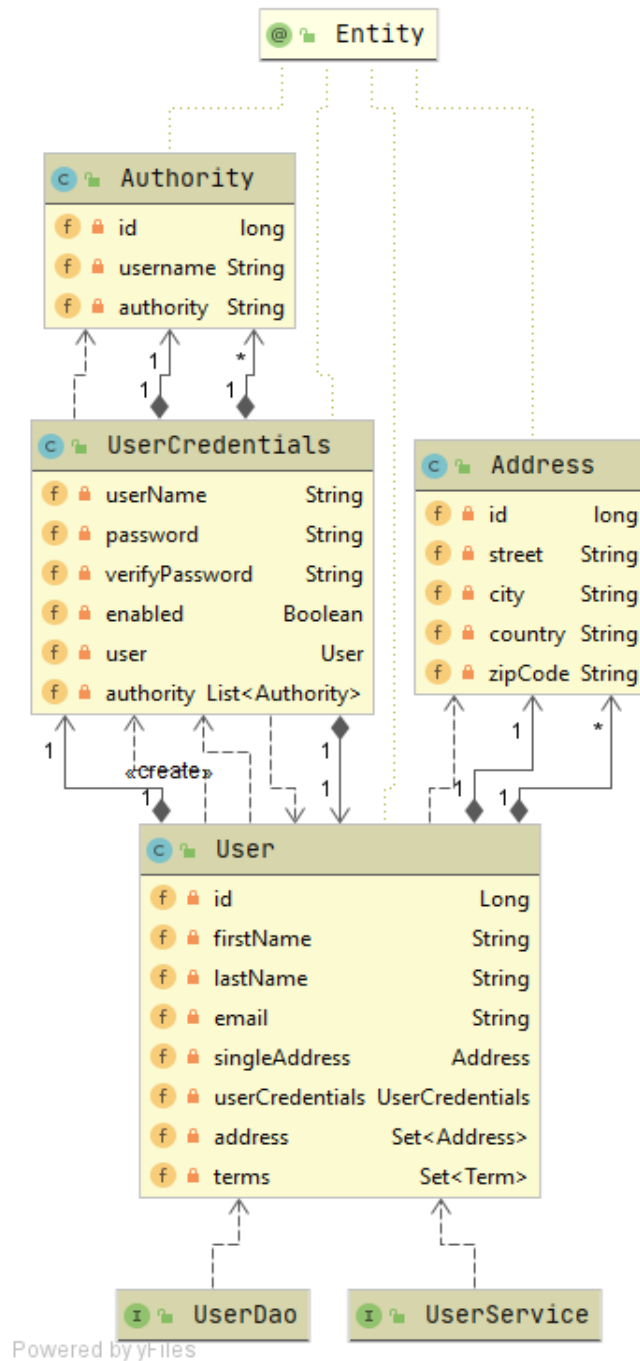


Figure 2. User Class Diagram

CAPTAINS - SlangDictionary

4.2 User Sequence Diagram

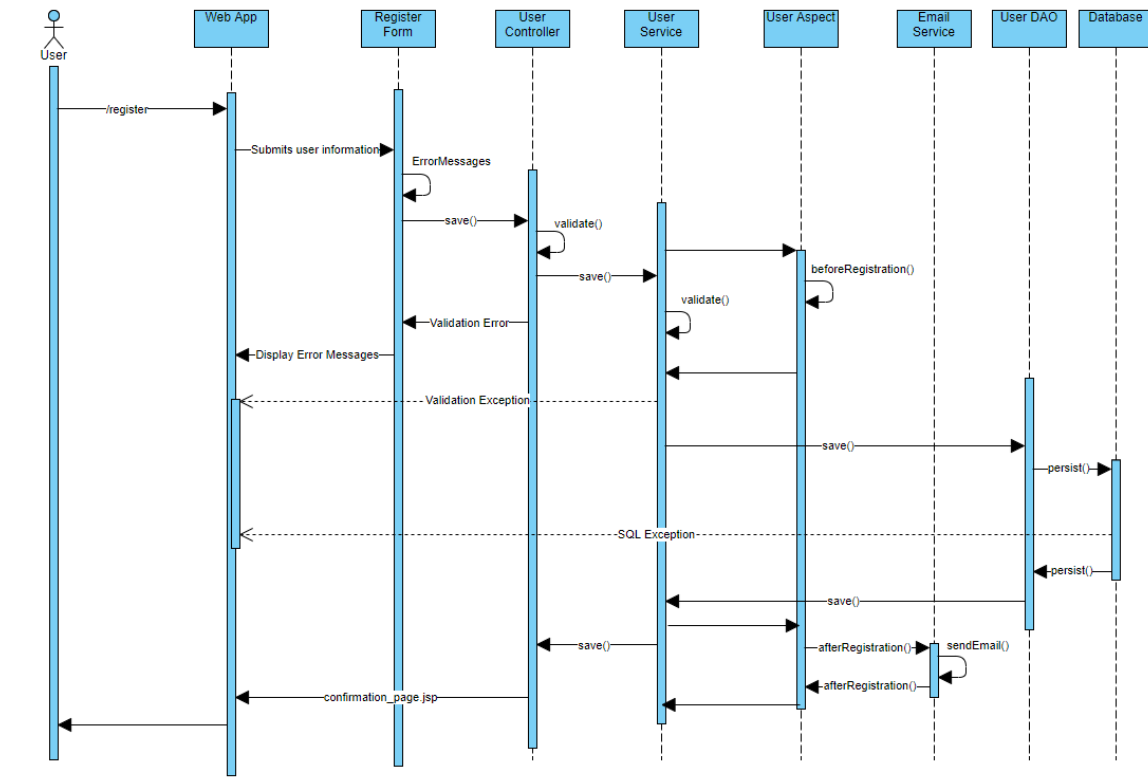


Figure 3. Sequence Diagram User Module

The user submits data to the register path, the request is then executed in the User Controller, data is validated through the validation and if there is any error the controller redirects the user to the registration page together with the error messages, if there is no error user save is called in User Service but before the save execution the User Aspect is called to modify some user data, then the validation is invoked one more time on the user domain to make sure everything is as it was in the first time the validation happened if there is an error there will be an exception otherwise the user save proceeds to the user dao by invoking the persist() to persist data on the database, if there is no SQL Exception and the data is persisted into the database, User Aspect is invoked by calling the after

CAPTAINS - SlangDictionary

registration method and proceeding to the Email Service, and finally the controller will return the confirmation page for successful registration.

4.3 Term List Sequence Diagram

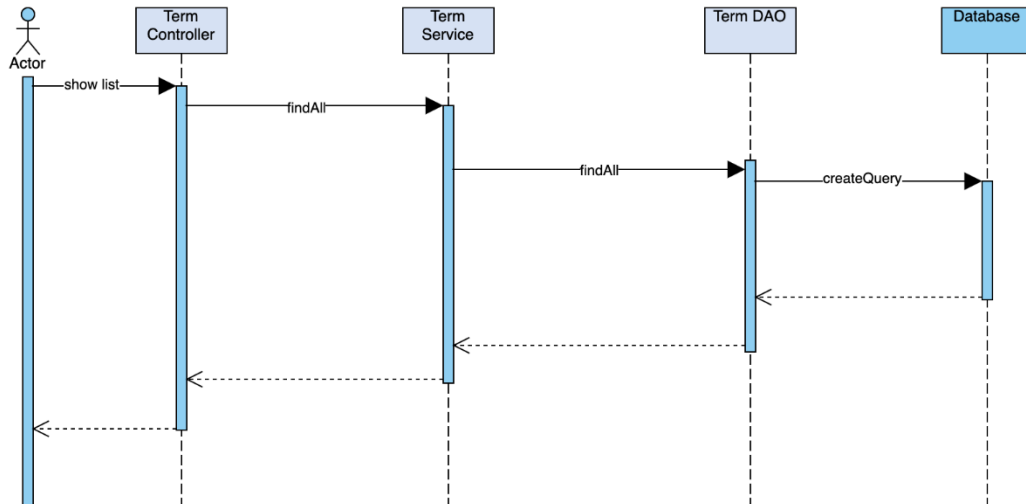


Figure 4. Sequence Diagram TermList Module

CAPTAINS - SlangDictionary

4.4 Add Term Sequence Diagram

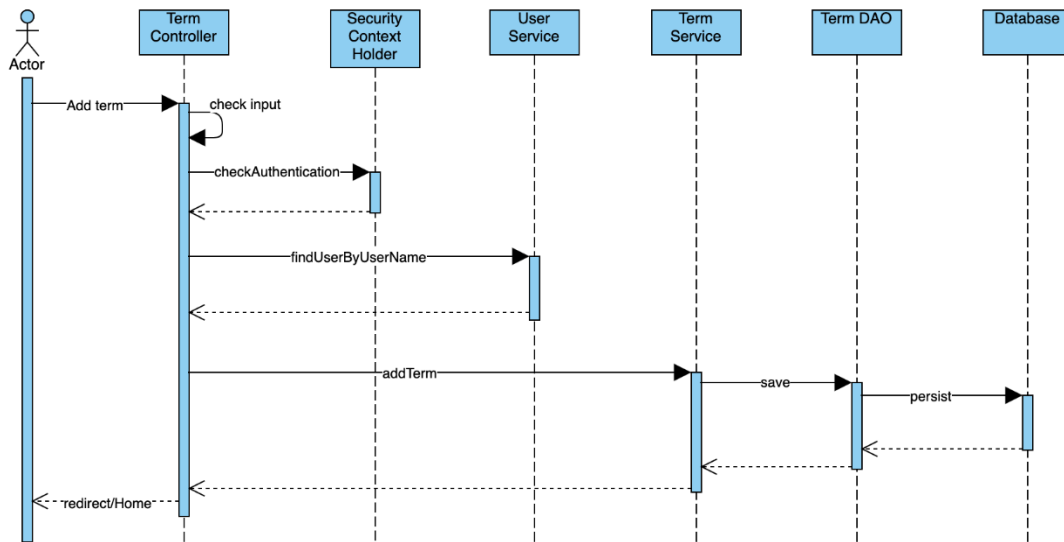


Figure 5. Sequence Diagram AddTerm Module

4.5 Category Sequence Diagram

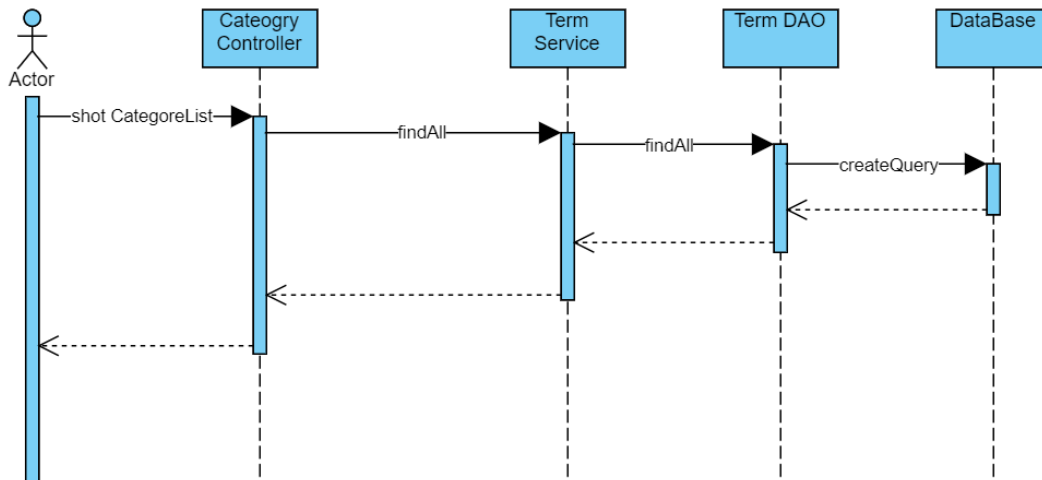


Figure 6. Sequence Diagram CategoryList Module

CAPTAINS - SlangDictionary

5. Design Alternatives

There is an opportunity to develop using the Spring MVC already have studied in the lesson.

6. Issues, Risk and Dependencies

- The error message cannot be displayed when a new user has already registered on the system
- The system cannot be automated when users enter a new word and need to enter a new category

7. Future Considerations

- Include a brief description with picture for each word,
- Include links to words with similar meanings,
- Depending on the size of the data to organize on the microservice.

8. References

<https://online.visual-paradigm.com/app/diagrams>

<https://www.urbandictionary.com/>