Design Document
Lyst: Recipe

Tristan Dang
Cassiee Latshaw
Eddy Rodriguez
Dilip Shah

March 31, 2017

TABLE OF CONTENTS

1. Introduction 2
1.1 Purpose 2
1.2 Project Scope 2
1.3 Document Conventions 2
1.4 Intended Audience 2
1.5 References 2
2. Architecture [PR1] 3
2.1 Product Perspective 3
2.2 Relationship Diagram [PR4, PR5] 3
2.3 Use Case Diagram [PR5, PR6] 4
2.4 Class Diagram [PR3, PR4, PR6, PR9] 4
2.5 Package Diagram [PR3, PR4, PR6, PR9] 5
2.6 State Diagram [PR6] 5
2.6.1 Search Recipe States 5
3. Use Case Walkthrough [PR2] 6
3.1 Recipe Search 6
3.2 Register User 7
4. Supporting Information [PR7, PR8, PR10] 8
4.1 Quality Attribute Design Tradeoffs 8
4.1.1 Usability 8
4.1.2 Conceptual Integrity 8
4.1.3 Performance 8
4.1.4 Security 8
4.1.5 Scalability 8
4.2 Architecture Approach Considerations [PR8] 8
5. Pseudo Code [PR1, PR3] 9
5.1 Search Recipe 9
5.2 Register 9
5.3 Login 10
5.4 Mark Recipe as Trusted 10

1. INTRODUCTION

1.1 PURPOSE

This project will result in the development of the Lyst: Recipe system, a system where users may enter a list of ingredients that they have available and will be given a list of recipes they could make with said ingredients. The user will select from abridged lists of ingredients to avoid validating user entry, and will be subject to several criteria such as food groups (meat, grain, vegetable, fruit, dairy), meal type (breakfast, lunch, dinner, appetizer, dessert), and required preparation time.

The Lyst: Recipe system is predicated on the concept of being a social network in that its content is heavily user-based. Recipes will initially be sourced by developers to ensure trusted content, but will eventually be more self-regulating by its users as the consumer base grows. This is done by allowing users to submit their own recipes as well as comment and rate existing ones. Recipes entered by developers from known sources (such as Food Network, allrecipes.com, or McCormick) will be marked as trusted, and user submissions will require a certain number of positive ratings to be marked as trusted in order to ensure a certain level of quality.

1.2 PROJECT SCOPE

The scope of the Lyst: Recipe project includes the planning, design, development, and testing of a prototype iteration of the system. The scope of this project also includes the completion of several deliverables including Software Requirements Specifications, Design Document, Test Procedures, and a User/Maintenance Guide. Project completion will occur at the end of the spring semester, after the prototype iteration of the system has been successfully completed.

For the prototype iteration, recipes in the database will be primarily sourced from hardcoding by the project development team. These recipe database entries will be automatically marked as trusted to serve as a starting point for the system's audience.

The Lyst: Recipe project will require the use of external webhosting and MySQL (My Structured Query Language) for database management, but no further outsourcing in terms of software or staff. Funding will not be required.

1.3 DOCUMENT CONVENTIONS

This document follows standard MLA Format. Bold-faced and capitalized text is used to indicate section and sub-section headings. Italics are used to reference sections within the document or other documents related to the project. Tables are used as organizational tools to be quickly referenced.

1.4 INTENDED AUDIENCE

This document is intended to be read by the project team and Project Sponsor. It is organized in such a way that all the system design should be easily understood by all project team members including the Team Lead, Team Scribe, individual deliverable leads, and developers. This document should give all readers a clear understanding of the Lyst: Recipe system design in order to produce a successful system.

1.5 REFERENCES

This document should be used in conjunction with the *Project Management Plan* and *Software Requirements Specification* deliverables created earlier during the spring semester.

2. ARCHITECTURE [PR1]

2.1 PRODUCT PERSPECTIVE

This system shall have a web-interface that allows users to interact with the database. Users will be able to create an account in order to post recipes, rate recipes, and comment on recipes. The following descriptions serve as brief summaries of the system's functions in accordance with the requirements listed in the *Software Requirements Specification* deliverable.

On the home page, users (both non-registered and registered users) shall be able to search for recipes. The user shall be able to search the database according to their selections and will be presented with a page displaying the recipes they can make. The user will be able to click on the recipe they wish to see in detail.

Upon clicking on a recipe, the user will be presented with a page displaying said recipe's name, description, ingredients, meal type, preparation time, instructions, rating, and comments. The user shall be able to rate the recipe according to a 1-to-5 scale and comment on the recipe using a text entry.

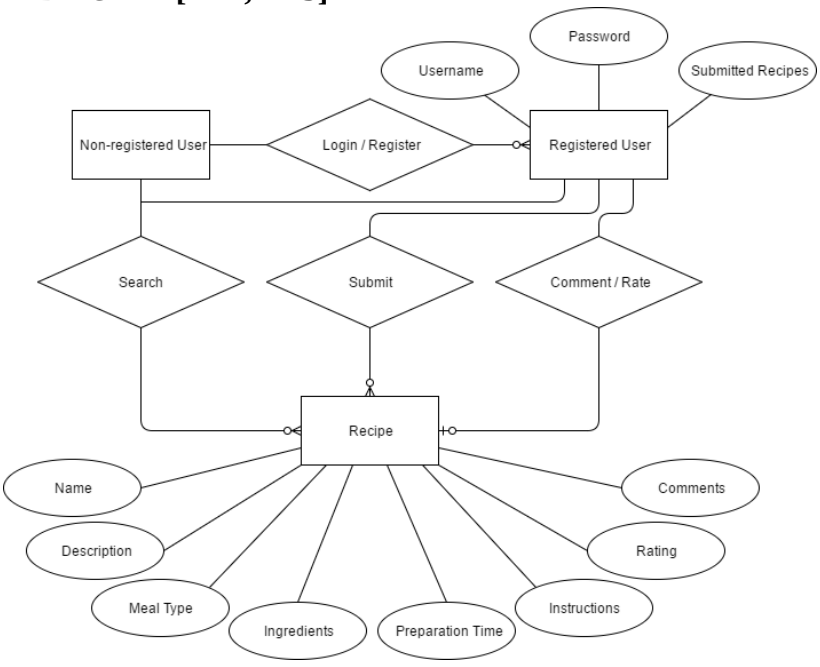
In the banner of the web-interface, there will be a login button. Upon clicking this button, users will be sent to a page where they may login or register.

If the user is logged in, there will be a submit recipe button in the banner of the web-interface. Upon clicking this button, users will be sent to a page where they may submit their own recipes by entering in the required information.

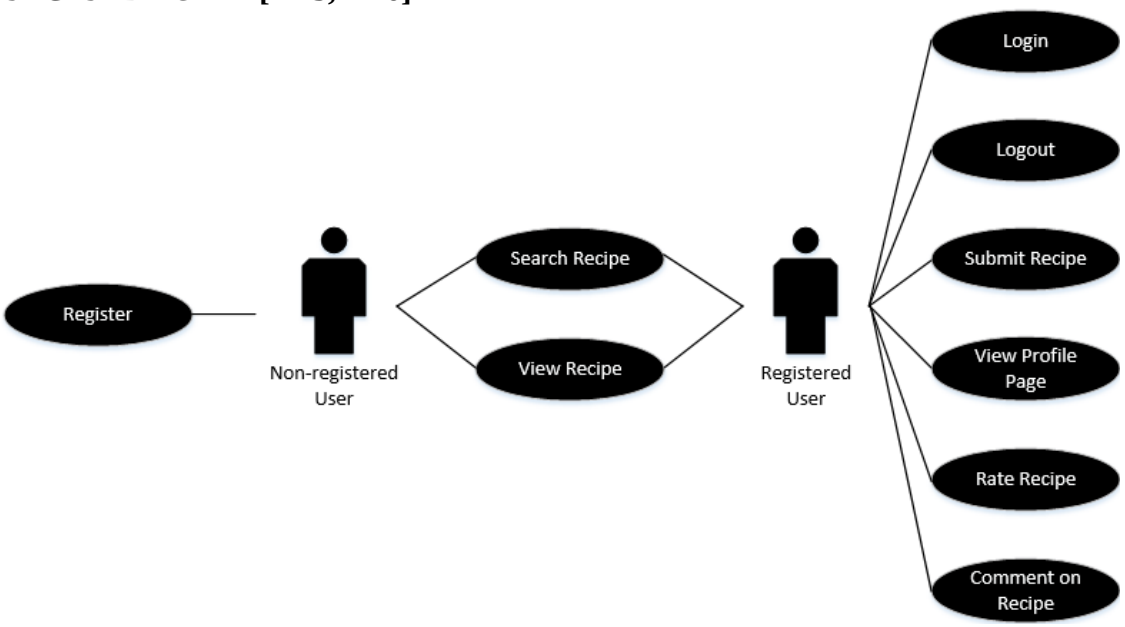
If the user is logged in, there will be a profile button in the banner of the web-interface. Upon clicking this button, users will be sent to a page displaying their profile name and list of submitted recipes. The user will be able to click on the recipe they wish to see in detail as described above.

If the user is logged in, there will be a logout button.

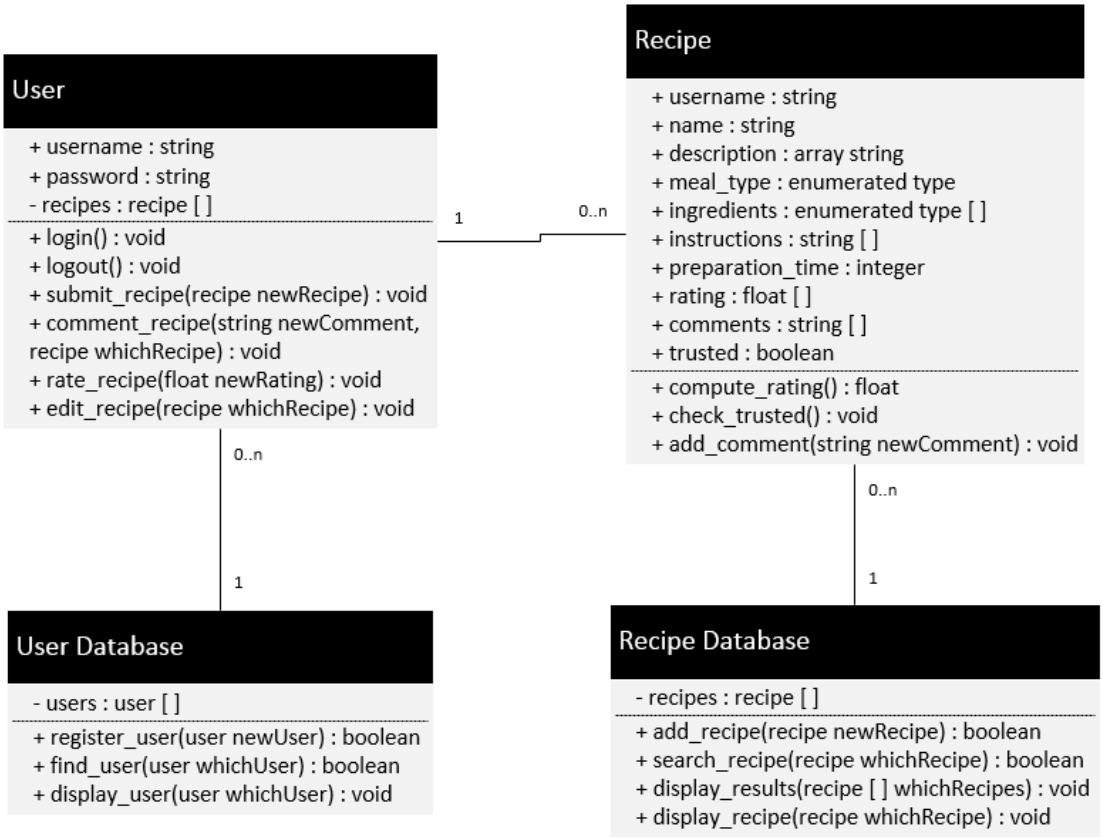
2.2 RELATIONSHIP DIAGRAM [PR4, PR5]



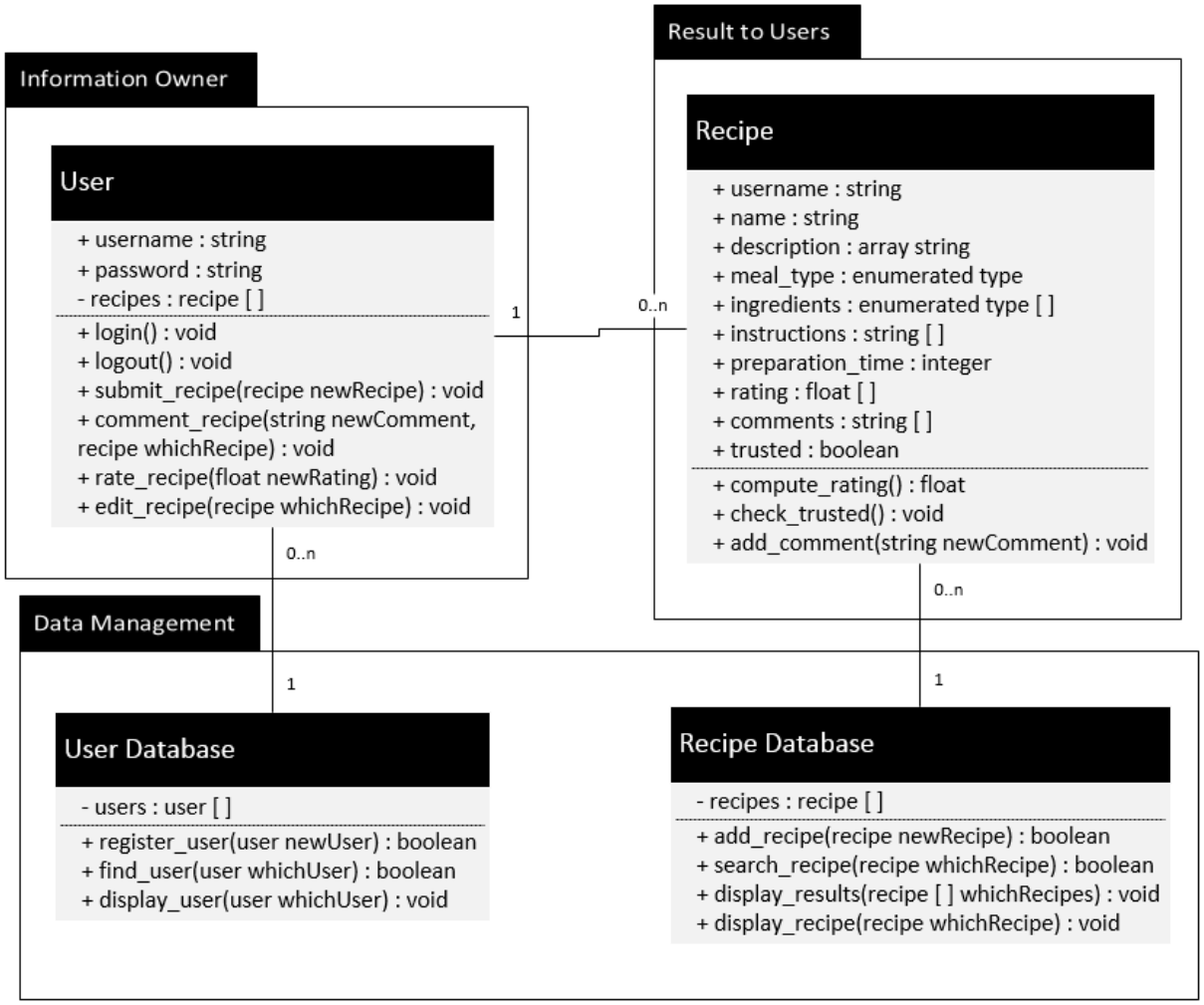
2.3 USE CASE DIAGRAM [PR5, PR6]



2.4 CLASS DIAGRAM [PR3, PR4, PR6, PR9]

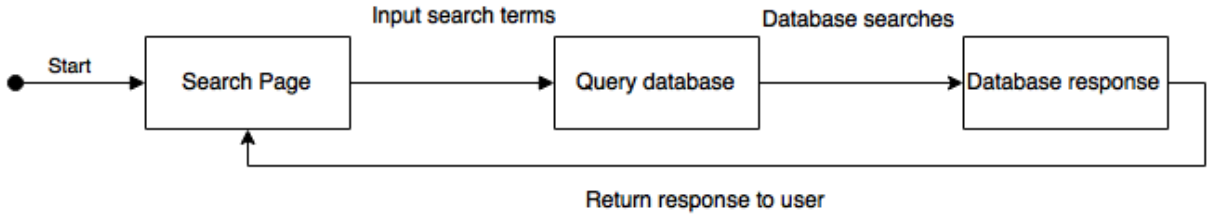


2.5 PACKAGE DIAGRAM [PR3, PR4, PR6, PR9]



2.6 STATE DIAGRAM [PR6]

2.6.1 SEARCH RECIPE STATES



3. USE CASE WALKTHROUGH [PR2]

3.1 RECIPE SEARCH

Use Case ID	RS
Primary Actor	All system users
Secondary Actor(s)	N/A
Description	This use case describes the process of searching the recipe database and viewing the list of recipe results based on the user's inputs.
Goal	To give users a visual representation of all recipes that satisfy their search query.
Success Management	The system successfully displays a list of recipes on the screen and users are able to successfully view each recipe and return to the search result.
Scope	Data
Precondition(s)	The user must be viewing the system's home page.
Typical Flow of Events	1. The user selects an ingredient from any of the food group drop-down menus. 2. After selecting an ingredient, the systems displays another drop-down menu from that food-group underneath the selected ingredient. 3. The user may select as many ingredients as they want as long as they include at least five ingredients in their query. 4. The user selects a meal type from the meal type drop-down menu. 5. The user selects a maximum preparation time from the maximum preparation time drop-down menu. 6. If the user has selected at least 5 ingredients, a meal type, and a maximum preparation time, the system displays a "Search" button underneath all of the input fields. 7. The user clicks on the "Search" button. 8. The system checks each recipe in the database to see if it satisfies all of the query's criteria. 9. If a recipe satisfies all of the criteria, the system adds the recipe to the list of search results. 10. The system displays all of the recipes that satisfy the criteria. 11. The user may navigate back and forth from any recipe in the list and the recipe results.

3.2 REGISTER USER

Use Case ID	RU
Primary Actor	Non-registered user
Secondary Actor(s)	N/A
Description	This use case describes the process of registering a new account for a user.
Goal	To allow users to register and create a Lyst account.
Success Management	The system successfully creates a user entry with the user's credential and displays the registered user's profile page.
Scope	Data
Precondition(s)	The user must be viewing the system's Login/Register page.
Typical Flow of Events	1. The user enters their desired username and password into the "Username" and "Password" input fields, respectively. 2. The user re-enters their password into the "Password Confirmation" input field. 3. If the user has entered credentials into all input fields, the system displays a "Register" button underneath all of the input fields. 4. The user clicks on the "Register" button. 5. The system checks each user in the user database to see if there already exists a user with the entered username. 6. If a user with that username already exists in the user database, the system displays an error saying "Username already exists" and returns the user to the Login/Register page. 7. If there is no user with the entered username in the user database, the system checks to see if the "Password" and "Password Confirmation" fields match. 8. If the two password fields don't match, the system displays an error saying "Password and Password Confirmation fields must match" and returns the user to the Login/Register page. 9. If the two password fields match, the system creates a new user entry with the entered username and password. 10. Then the system displays the registered user's profile page and a confirmation saying "Registration complete."

4. SUPPORTING INFORMATION [PR7, PR8, PR10]

4.1 QUALITY ATTRIBUTE DESIGN TRADEOFFS

The following sections are quality attributes as agreed upon by the project team and Project Sponsor in the *Software Requirements Specification* deliverable. The sections describe the project team's plans in terms of design and respective quality attribute tradeoffs. The quality attributes are listed in order of priority, with higher sections denoting higher precedence in terms of tradeoff between attributes.

4.1.1 USABILITY

The Lyst: Recipe system is predicated on the concept of being a social networking application, thus users must be able to navigate the interface intuitively and efficiently. The project team will seek to maximize usability throughout development while adhering to the *Software Requirements Specification* deliverable. This will be accomplished through efficient and cleanly designed user interfaces. This quality attribute will take precedence over the ones listed below.

4.1.2 CONCEPTUAL INTEGRITY

The project team will seek to adhere to best coding practices during development with cohesive and efficient code and consistent comments, with project completion taking precedence. Best coding practices will be performed whenever possible, but if time restrictions become apparent, the project team will prioritize usability and completion.

4.1.3 PERFORMANCE

In order to maximize usability, the system must be able to respond in a timely manner when submitting the recipe search, logging in, adding a recipe, rating a recipe, and commenting on a recipe, with as little compromise as possible. Usability, in terms of interface and database setup, will take precedence over performance. Basic SQL statements produced by phpMyAdmin will be used to query the database with no alterations to their given performance.

4.1.4 SECURITY

Time restrictions due to the CSCE 4901 course timeline make it necessary to put system completion over high level security. The system will be able to handle basic login/registration functionalities but will not be encrypted at this stage of development. The security requirements specified in the *Software Requirements Specification* deliverable will take precedence over high level security.

4.1.5 SCALABILITY

Time restrictions due to the CSCE 4901 course timeline make it necessary to put system completion over long term scalability. There are no current plans to pass the system to another team following course completion. The requirements specified in the *Software Requirements Specification* deliverable will take precedence over long term scalability.

4.2 ARCHITECTURE APPROACH CONSIDERATIONS [PR8]

The project team has explored two main approaches for the system's architecture: AlterVista and WordPress. AlterVista is an Italian web platform that offers free webhosting in conjunction with phpMyAdmin and SQL database functionality. WordPress is a website creation tool that offers an array of open source plugins that include database functionalities. The project team will agree upon the approach that best allows for system completion in accordance with the *Project Management Plan* schedule.

Quality Attribute	AlterVista	WordPress
Usability	Italian based platform has less resources for users	Widely used open-source tool that offers many intuitive plugins for users
Conceptual Integrity	Resources and source code often require translation	Robust code often already produced by extensions
Performance	Hosts site directly, higher performance	Does not host site and requires external hosting, therefore lower performance
Security	Has a lower frequency of updates that may leave room for less security	Has a higher frequency of updates to improve security
Scalability	Offers less plugins for extensibility	Offers more plugins for extensibility

5. PSEUDO CODE [PR1, PR3]

5.1 SEARCH RECIPE

```
// Requirements Satisfied: R001, R002, R003, R004
search_recipe() {
        get input from user
        if input requirements met (i.e., at least 5 ingredients, a meal type, and a preparation time) {
                query database
                display results
        }

        else if input requirements not met {
                cout << “Please enter in all of the required fields.” << endl;
        }
}
```

5.2 REGISTER

```
// Requirements Satisfied: R006
register_user(desired_username, desired_password, password_confirmation) {
        if desired_username is already in use {
                cout << “Username already exists.” << endl;
        }

        else {
                if desired_password does not match password_confirmation {
                        cout << “Password and Password Confirmation fields must match.” << endl;
                }

                else {
                        display new user’s profile page
                        cout << “Registration complete.” << endl;
                }
        }
}
```

5.3 LOGIN

```
// Requirements Satisfied: R007
login(username, password) {
	if username doesn't exist {
		cout << "Username or password is incorrect." << endl;
	}
	else {
		if password doesn't match {
			cout << "Username or password is incorrect." << endl;
		}
		else {
			display user's profile page
		}
	}
}
```

5.4 MARK RECIPE AS TRUSTED

```
// Requirements Satisfied: R013
check_trusted() {
	rating_count = compute count of recipe's rating
	rating_average = compute average of recipe's ratings

	if(rating_average >= 4 && rating_count >= 5) {
		trusted = true
	}

	else {
		trusted = false
	}
}
```