Roommate Application

Date: 06 Dec 2016

Team Members:

Shrivathsa Murthy

Athreya Nakshathri

Chaithanya Rao

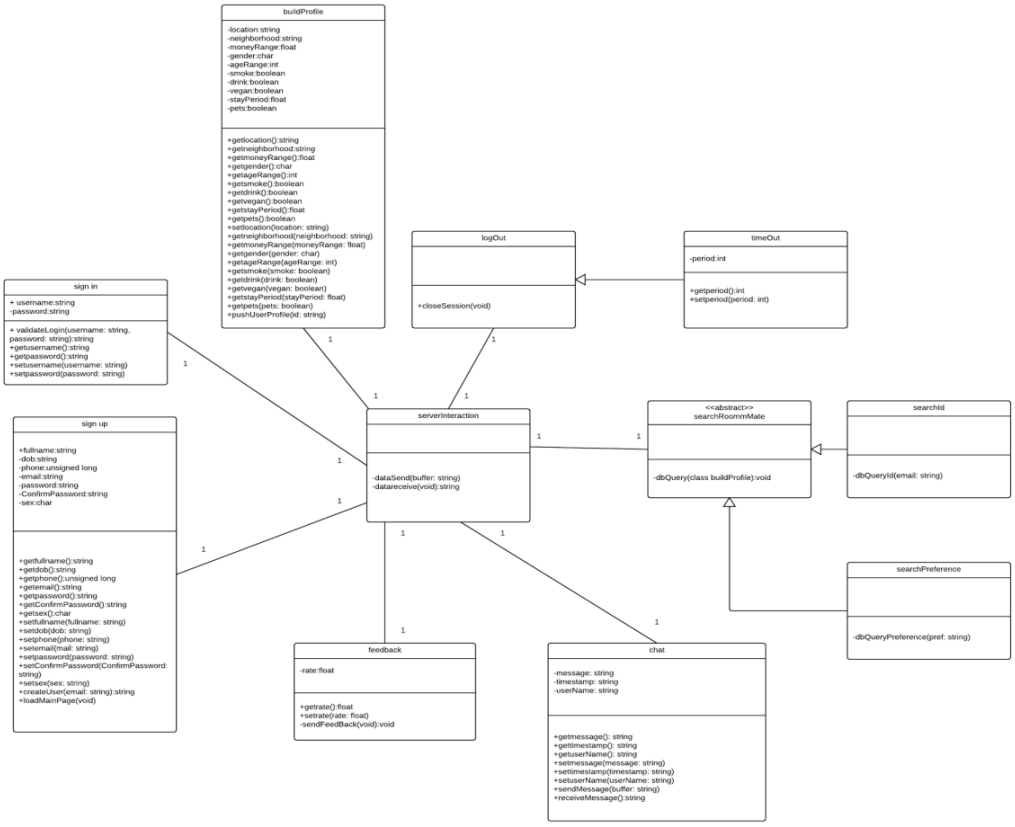
Implemented Use Cases and Features

Use Case ID	Title
UR-001	A user must be able to search for other roommates based on name or email-id
UR-006	A user must be able to edit his profile
UR-009	User must be able to get notifications about the matched roommates and the criteria based on matching
UR-010	User must be able to write a testimonial about the application
UR-011	New user/Existing user must be able to see the main menu after successful login
UR-012	Existing user must be able to see his saved profile
UR-013	User must be able to successfully log out after hitting the log-out button
UR-014	User must be able to click on FAQ to see more about the app's matching functionality
UR-015	User must be able to recover the password
UR-017	User must be able to contact Admin via Cust Care Number if the app is buggy

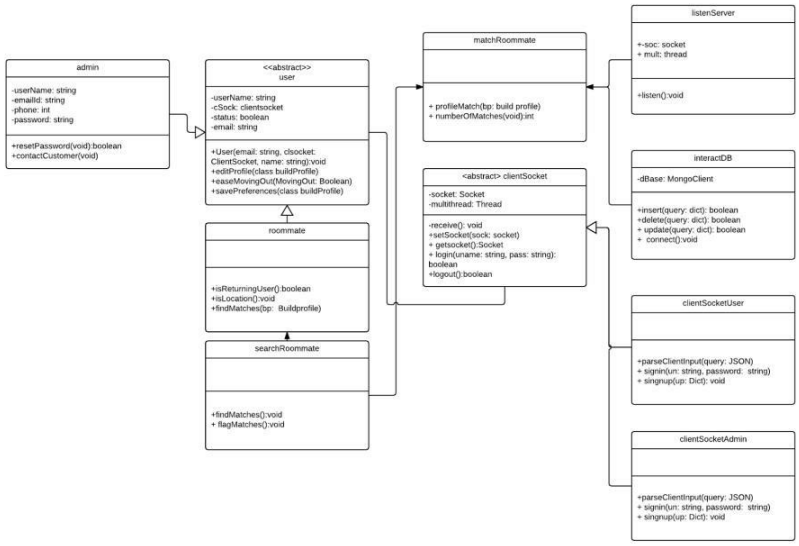
Use Cases Not Implemented

Use Case ID	Title
UR-002	Only upon accepting the recommendation/requests, a user must be able to chat with the accepted roommate
UR-003	Users must be able to decline/accept the recommendation
UR-004	User must be able to view a history of recommendations and searches
UR-005	Users must be categorized as one with the place or looking for a place
UR-007	When a roommate leaves the house, users must be able to rate each other
UR-008	A user must not be able to see his own rating, but others' rating must be visible
UR-016	User must be able to sign up using Facebook account

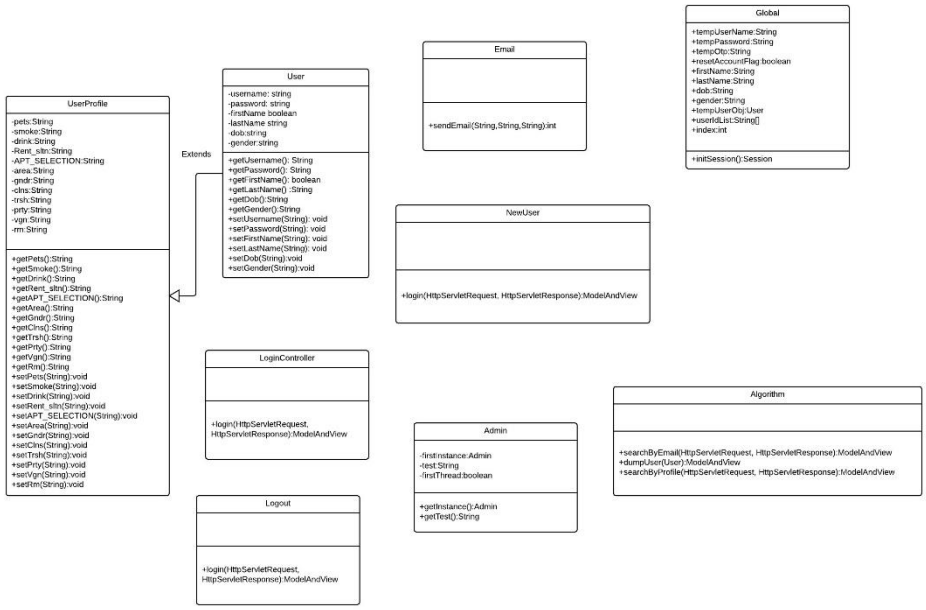
Project Part-2 Class Diagram (Client Side)



Project Part-2 Class Diagram (Server Side)



Project Part-3 Class Diagram (Server Side)

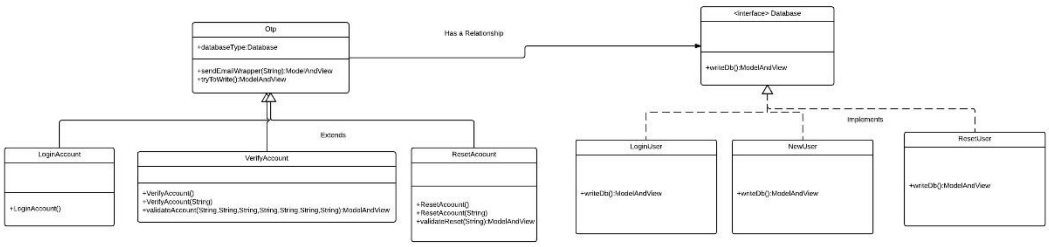


Did you make use of any design patterns in the implementation of your final prototype?

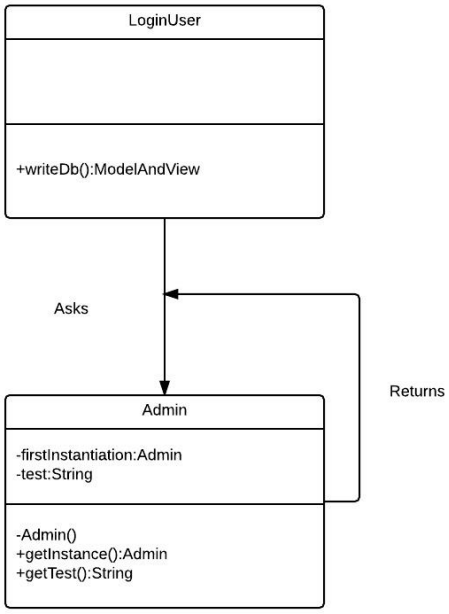
Yes. We have used Strategy, Singleton and Factory Design patterns.

If so, how? Show the classes from your class diagram that implement each design pattern (each design pattern as a separate image in the .PDF).

Project Part-3 Class Diagram (Strategy Pattern)

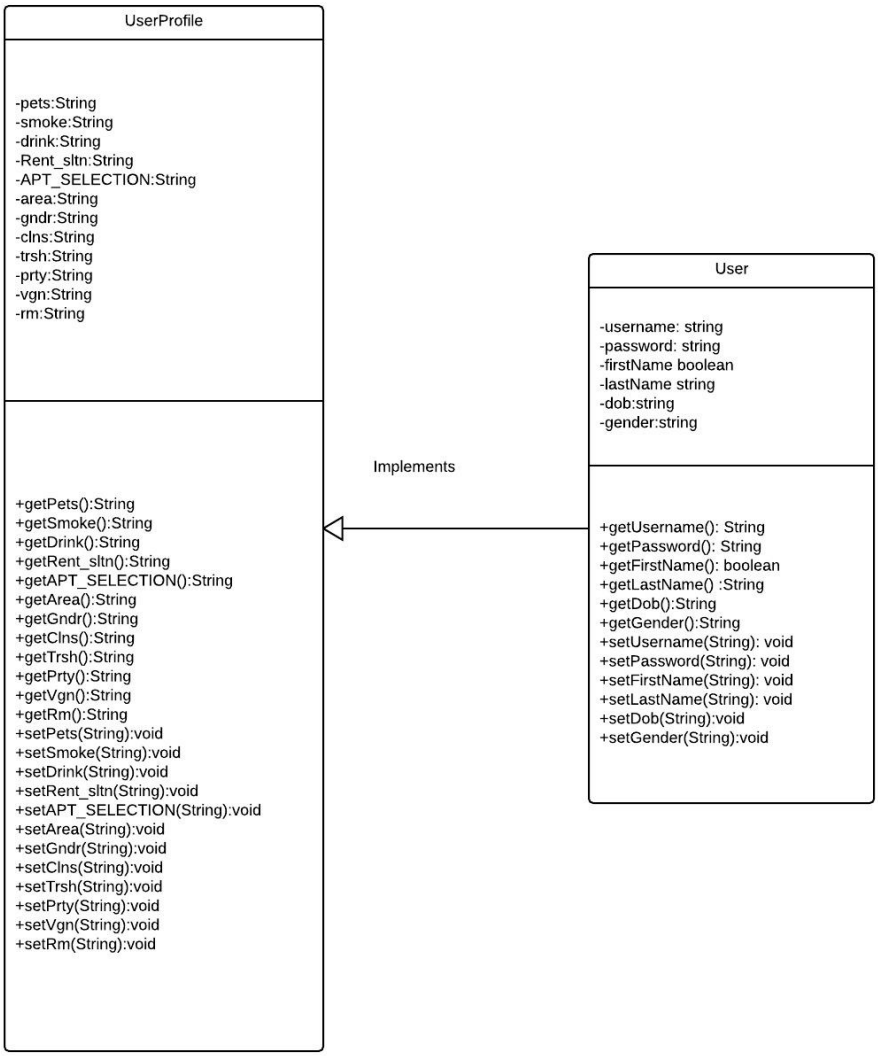


Project Part-3 Class Diagram (Singleton Pattern)

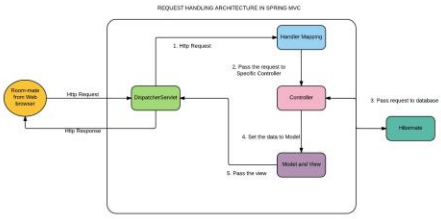


Project Part-3 Class Diagram (Factory Pattern)

HIBERNATE USES SESSION FACTORY DESIGN PATTERN
Here we are indirectly persisting UserProfile and directly persisting User



Project Part 3 – Spring MVC Design Architecture



If not, where could you make use of design patterns in your system? Show a class diagram of how you could implement each design pattern and compare how it would change from your current class diagram.

5. What have you learned about the process of analysis and design now that you have stepped through the process to create, design and implement a system?

We would have blindly started to code without the knowledge of Design Patterns in OOP. Now that we have a good exposure to that, we have realized that it is much easier to translate an existing UML diagram into code. It is always a better idea to spend more time in designing patterns which saves significant amount of time in implementing the same.

Having a version control is mandatory as it will eliminate debugging hassles.