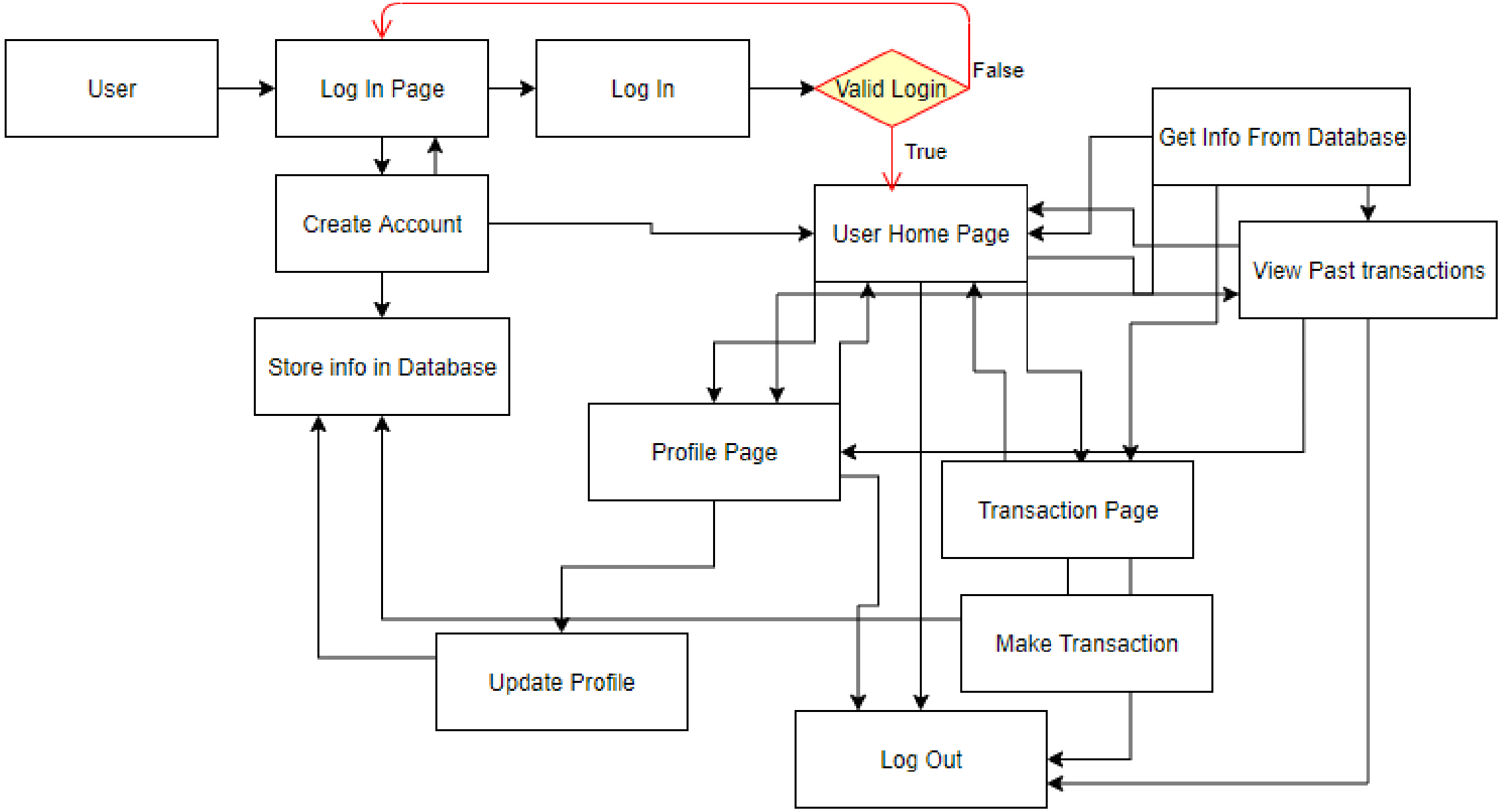
By Sean Grano

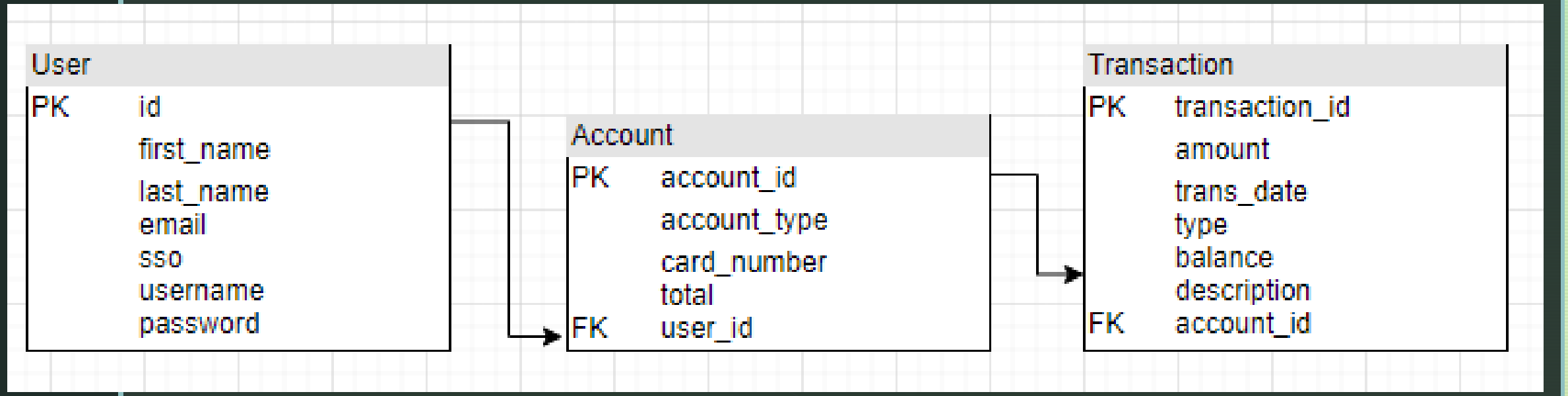
Online Banking Application

Tech Stack

- Java
- Sql
- Javascript
- Spring Boot
- .jsp
- html
- css
- bootstrap


User
Log In Page
Log In
Valid Login
False
True
Create Account
User Home Page
Get Info From Database
View Past transactions
Store info in Database
Profile Page
Transaction Page
Make Transaction
Update Profile
Log Out

Schema Diagram



Tasks

- 1. Create an HTML Log In page
- 2. Create an HTML Create Account page
- 3. Create an HTML View Profile Page
- 4. Create a HTML Transaction page
- 5. Create a HTML Past Transactions page
- 6. Create CSS page and add inline CSS to pages as needed

Tasks cont.

- 7. Add javascript to Create Account page for password
- 8. Add javascript to Profile page to make it so User can only change one property at a time
- 9. Add javascript to Transaction page to ensure user inputs valid amounts
- 10. Convert HTML pages to .jsp pages
- 11. Create SQL Tables (Entities) for bank application
- 12. Create Spring Boot Bank Application

Tasks cont.

- 13. Create DTO classes that match up to the SQL tables
- 14. Create the DAO interfaces for the Entities
- 15. Create Transaction, Account and User Services
- 16. Create a controller class for viewing the pages
- 17. Create a controller class to add functionality to the pages
- 18. Connect to database through application.properties file and have spring boot read .jsp files

Tasks cont.

- 19. Create a method for each .jsp page in the ViewController page
- 20. Add methods to UserController that add functionality to the pages and add required html to .jsp pages
- 21. Create testing methods for each dao method used
- 22. Add exception handling
- 23. Add BCrypt to password and update login validation
- 24. Information displayed on the pages is taken from the database
- 25. Information entered into the pages is stored in the database

Challenges

- Problem: Designing the pages, deciding how each page should look
- Solution: Checked out some Banks websites and based mine off theirs
- Problem: Deciding what entities are needed for the bank application
- Solution: Looked up what information was displayed on a bank application and what information what required to create a bank application Then put them in groups and created tables based off those groups

Challenges cont.

- Problem: Getting the update function to work on the Profile page
- Solution: Used @ModelAttribute for updating and then checked which properties weren't null and then updated that property
- Problem: Getting the date for past transactions to appear without the time
- Solution: Used SimpleDateFormat to make it appear correctly