Ronald A. Canales
Jeremy R. Easton-Marks
Thi Quy T. Tran

Group 4 - Assignment 1

Initial Thoughts (TT)

- User experience & functionalities
    - Volunteers and Administrators (Administrators only: **BLUE**, both: **BLACK**):
        - Login page requires username (registered email) and password to access user account
        - Register page (new account registration) requires:
            - Account information: username (email - requires email verification), password
            - Personal information : full name and birthdate - requires ID verification
            - Contacts: phone numbers, email - requires email verification
            - Addresses: physical and mailing addresses
            - A user ID will be shown on the registration confirmation page
        - User account:
            - Profile (users can see, manage, update, or change):
                - User ID (UNIQUE AND CANNOT BE CHANGED)
                - Account information: username (email - requires email verification), password - hidden
                - Personal information: Full name and birthdate - changes requires ID verification
                - Contacts: phone numbers, email - requires email verification
                - Addresses: physical and mailing addresses (physical address will be set as default location of user)
                - Skills and preferences
                - Current availability status
            - Events/Tasks:
                - Completed and uncompleted events/tasks
                - Date, time, location of events/tasks
            - Events/Tasks management:
                - Creates new events/tasks and event assignment
                - Sets up and manages events/tasks' required skills
                - Sets up, updates and changes date, time, and location of events/tasks
                - Updates and changes completion status of specific volunteer in a particular event/task
                - Assigns specific volunteers to events/tasks by using volunteer matching (finds volunteers having event/task's required skills)

        - Manages volunteers attendance (check-in a volunteer by using volunteer's user ID and password that volunteer use to access user account) in a particular event/task
    - Notifications:
        - Event assignments, updates, and reminders
        - Creates and sends event assignments, updates, and reminders to volunteers
        - Users can see and search for administrators and volunteers' information: user ID, skills, preferences, availability status, full name, contacts (searching by using email/user ID/or name)
        - Sends messages to a specific Administrator/Volunteer
    - Volunteer History:
        - Volunteer participation history and performance
        - Rates overall and specific event/task of a particular volunteer's performance with feedbacks
- Technologies:
    1. Front end: React (https://react.dev/)
    2. Back end: Java Spring (https://spring.io/) (https://www.baeldung.com/building-a-restful-web-service-with-spring-and-java-based-configuration)
    3. Database: MySQL (https://www.mysql.com/)
    4. Code Repository: GitHub (https://github.com/JREastonMarks/COSC6353)

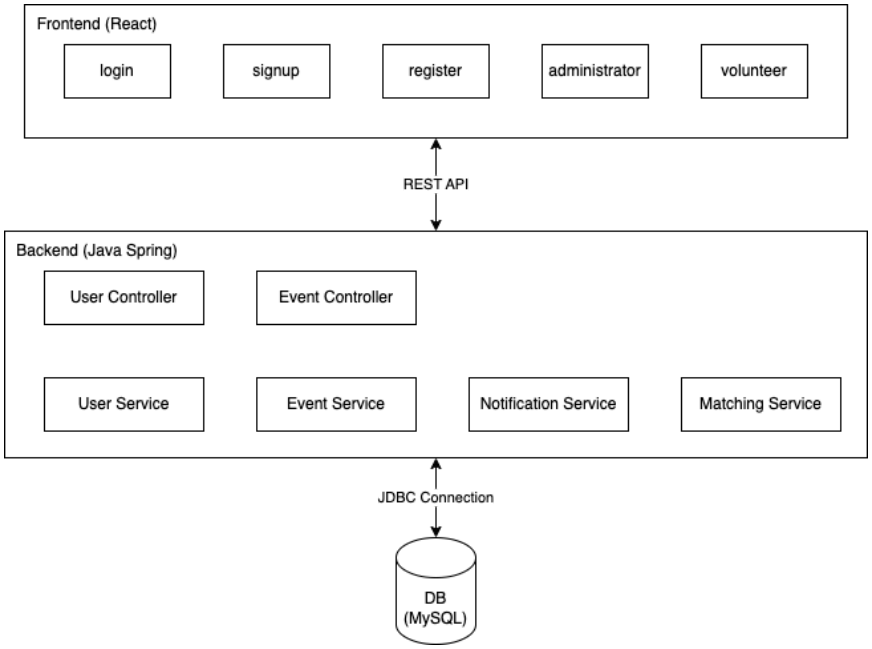
Development Methodology (RC)

- Agile with Test Driven Development: Since Agile development is intended to develop projects quickly with little time spent on design, it will be useful since the summer semester is shorter than a regular semester. In the class lecture, the professor spoke about how change was inevitable in software design, so using an Agile process will help the group mitigate unexpected changes in the given amount of time. By following the functionalities above and given the provided information flow charts below, we have detailed how each entity will interact with each other. This will help with developing tests to ensure our program works.
- With Test Driven Development, we will develop test cases that our program has to pass. As a result, the group will write the minimum amount of code to pass the tests, saving time and effort. Finally, we make our code ready for users and look at feedback for improvements or bug fixes. We will repeat the process until all test cases pass or project goals are achieved. We will also mix in some Scrum methods like short meetings and testing as we complete parts of the project to make sure everyone is following the schedule set and to solve any issues.

High-Level Design / Architecture (JEM)

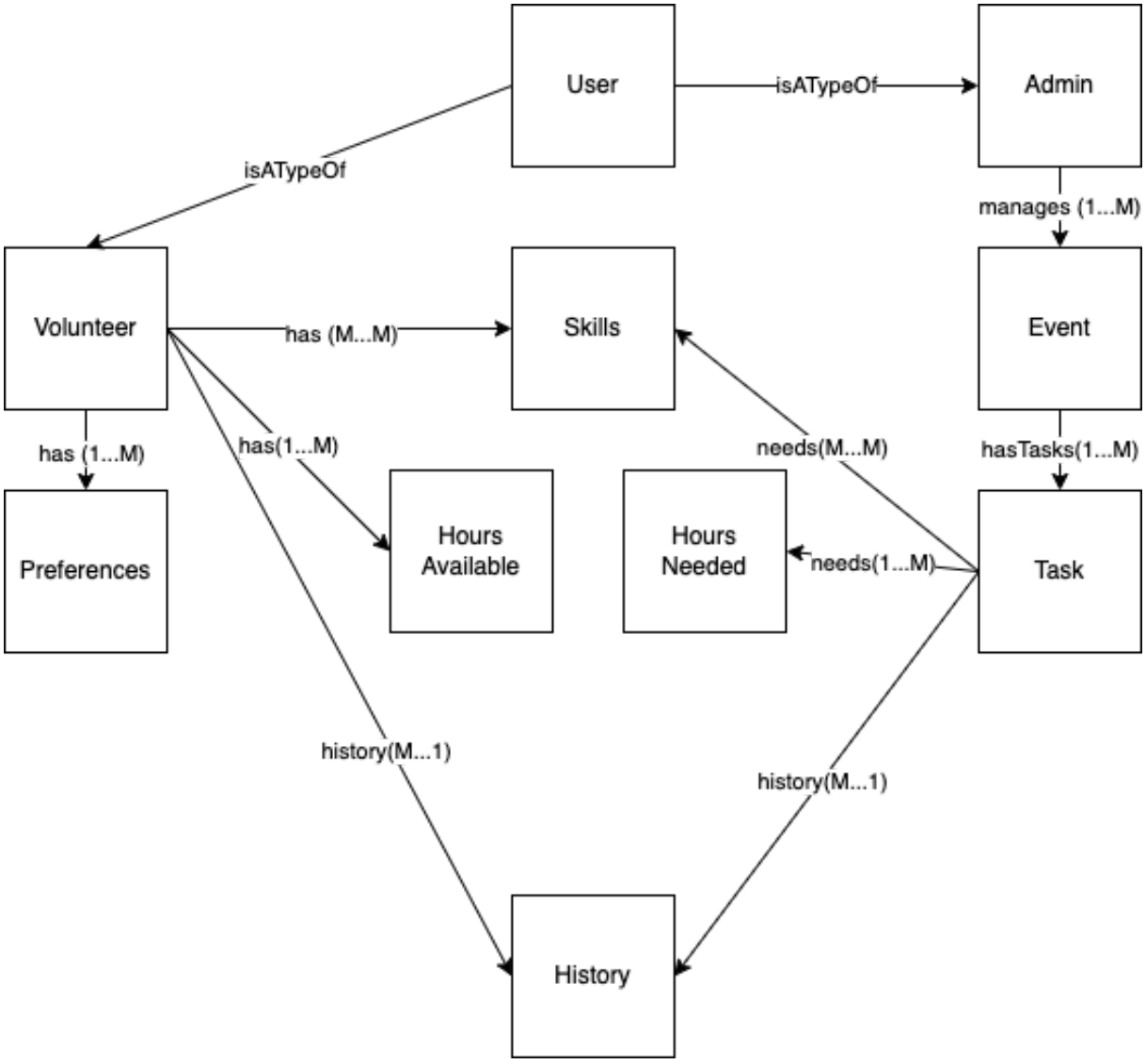
System Components Overview

As described above we identified three major layers to our application. The first being the frontend, that will be developed using the React Framework. This will have five major views; login, signup, register, administer, volunteer. We chose React as it is a well documented and popular web framework that will allow us to quickly create and iterate in the building of the frontend components. This layer will communicate with our backend using a REST API. The backend will be built using Java Spring. This will include two REST controllers for managing the users and events. We have identified four services that will need to be created; User, Event, Notification, and Matching. We have chosen Java Spring as it is a mature framework that contains components for easily handling security, testing, REST, and DB connectivity. This layer will communicate with the database using JDBC. We will store all data in a MySQL database as it meets our requirements for storing relational data.



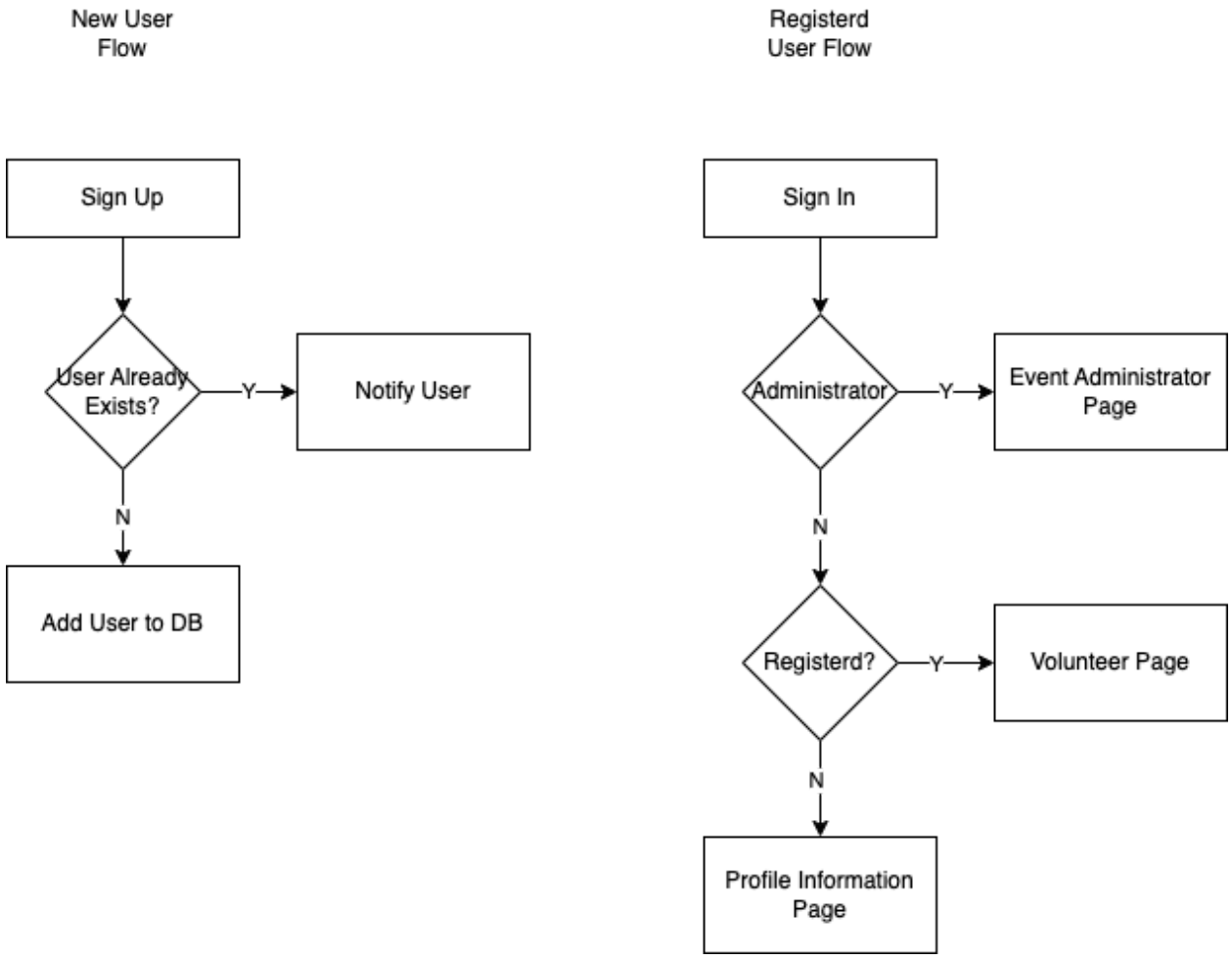
Object Model

We have divided our object model into three major areas. The first being the volunteer information which is a type of User. The volunteer object can have any number of skills and those skills can be shared amongst different volunteers. Additionally, each volunteer has a number of hours that are available and preferences. Our second area is the administrators that are also a type of User. They manage any number of events. Events is our third, and final major area. In which an event has any number of tasks which require specific skills, hours that those skills are needed and a history which links events and volunteers.



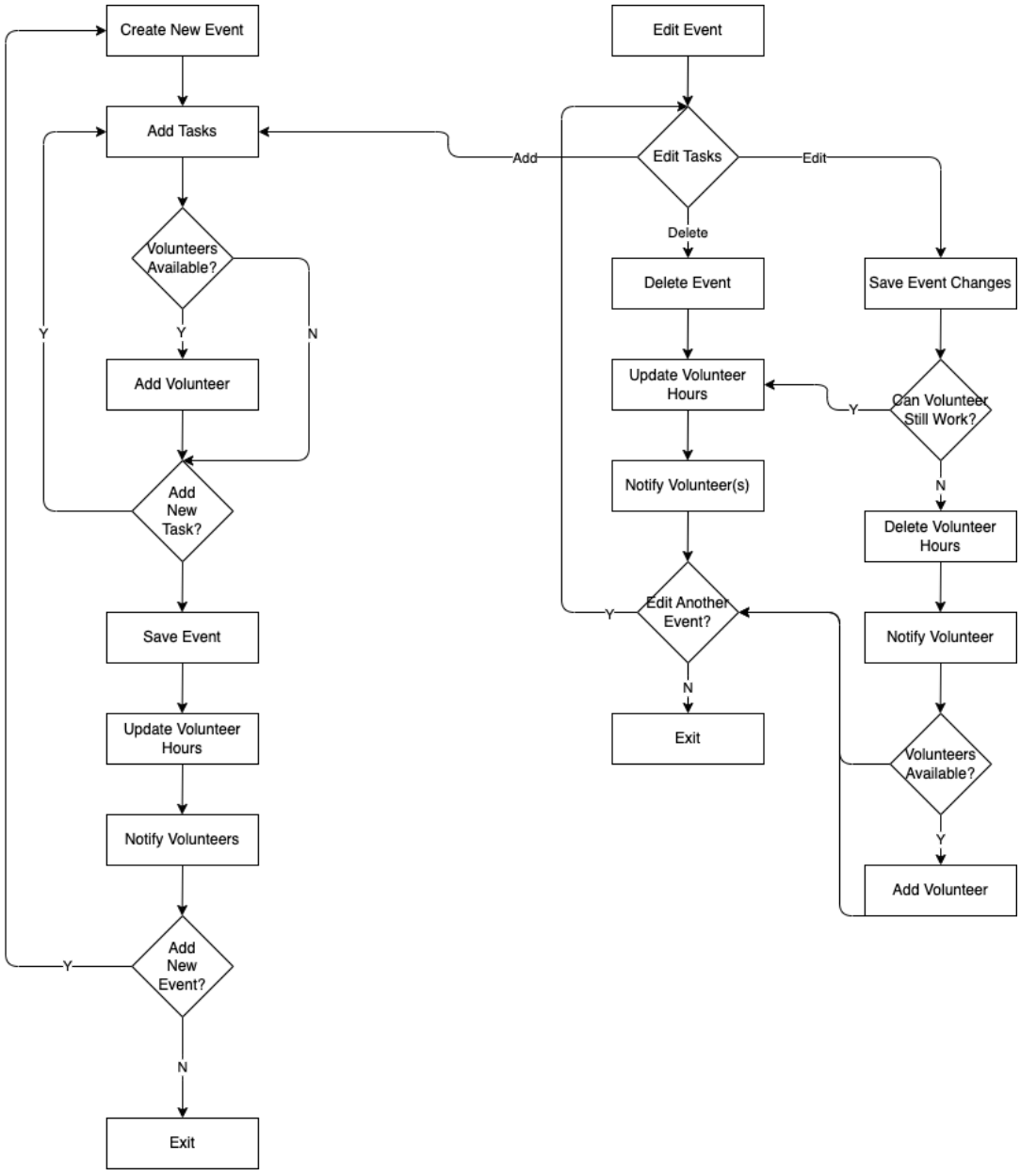
New User and User Registration Flow

Signing up users is a simple process in which we receive information about them signing up and add the user if they do not already exist in the database. Once a user is registered and they sign in what happens next depends on two things. The first being if they are an administrator or not. The second being that if they aren't an administrator then we need to know if they have registered their profile information.



Event Flow

We have identified two different flows in our application when it comes to events. The first is the creation of a new event and the second is the managing and editing of the event after it has been created. Both flows share similar steps such as saving/updating the data, notifying the volunteer when a change happens, and finding new volunteers. This will allow us to reuse these components in multiple places.



Member Contribution

Member Name	Contribution	Notes
Ronald A. Canales	Development Methodology	All members contributed equally
Jeremy R. Easton-Marks	Worked with team to create flows	All members contributed equally
Thi Quy T. Tran	Worked with team to develop initial thoughts of user experience and functionalities, also decided technologies for creating a website.	All members contributed equally