Analysis Me

**Spring**

2022

Technical Report

Team Members: Maria Harrison, Amayrani Luna, Blake Miller, Landon Wood

Instructor: Professor Jia Zhang

Enclosed in this document is the technical report of CS4345 Software Engineering Principles.

Computer Science Department

Southern Methodist University

# Table of Contents

1. **Introduction**
2. **Motivation**
3. **Related work**
4. **Design**
5. **System implementation**
6. **Experiments/Analysis**
7. **Known issues, bugs, problems**
8. **Conclusions and recommended future work**
9. **Documentation**

**9.1 Access Information**

**9.2 Download and Installation**

**9.3 Assumptions and Considerations**

**9.4 Tutorial**

1. **Introduction**For this semester project, our class was tasked with creating a sort of social network for scientists – think of LinkedIn or Twitter but targeted for those in academia. We defined 12 functionalities that we wanted our platform to allow at the start of the semester:   
     
    - Search information, e.g., which papers to read, who are experts in a topic, what’s the trend for a direction, who to follow, etc.  
    **-** Find collaborators for a research project  
    - Promote one’s paper publication in the society  
    - Form a virtual team to collaborate  
    - Follow a researcher; manage followers; tweet/retweet  
    - Follow a paper or study  
    - Ask a question and hope to receive discussions and answers  
    - Find someone to help with payment (freelancer style, e.g. publish a job)   
    - Build up personal profiles, including research interests, credentials, publications, projects, etc.  
   - Collaboratively design a data analytics experiment  
   - Community users can comment on a paper published  
   - Community users can link studies together to provide peer review.
2. Each team in our class selected a function that they wanted to work towards implementing. Our team, team 2, chose to work on the function “Build up personal profiles, including research interests, credentials, publications, projects, etc.” We worked on making a profile page throughout the semester.
3. **Motivation**

The motivation for creating a scientist social network is simple: we want professors, students, and peers to be able to connect and share interests/information among themselves. Many social media sites already exist where users can do so. However, we believe that a site specifically designed to connect academic figures would be better at fostering discussions and learning.

1. **Related Work**

Technically, any social media could be considered related work in this project domain. Scientists already communicate and collaborate on established platforms like LinkedIn, Twitter, etc. However, there are a couple more specific academically oriented social media sites: ResearchGate.net and Academia.edu. Both of these platforms allow users to upload and download scientific research articles. They allow users to connect and share information and receive support from each other. Sites like these are close to what we are trying to accomplish with the Scientist Social Network project.

1. **Design**For our profile page, we created a 4+1 view of our project during the design stage. Unfortunately, not all of our designs made it into the final submission, but below we will list our UML diagrams that show our vision for the project.

UML Class Diagram

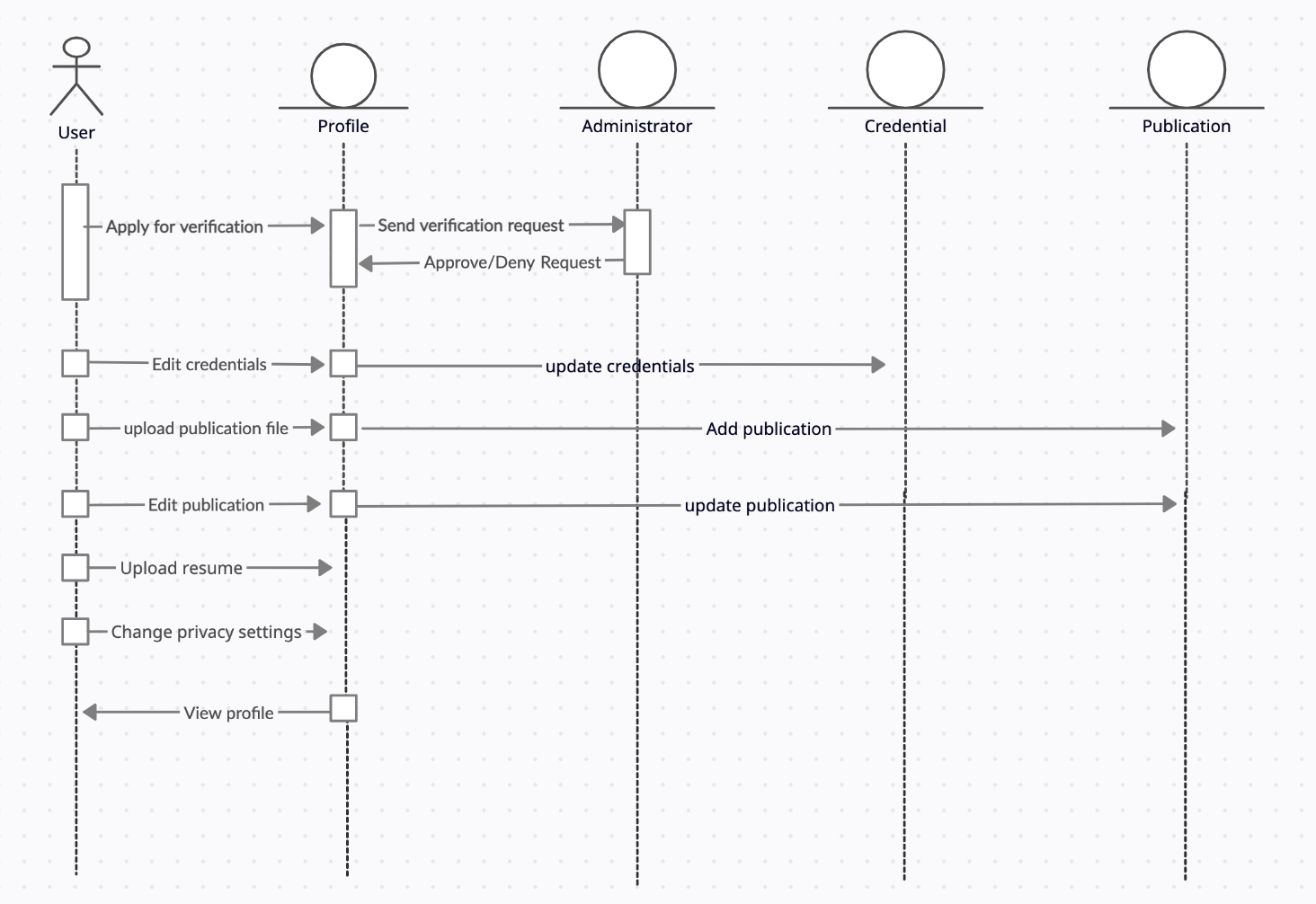
Diagram

Description automatically generated  
  
UML Development Diagram

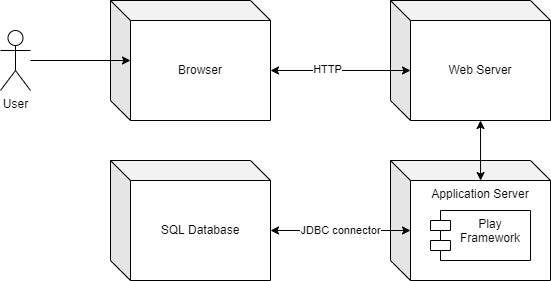
Diagram

Description automatically generated

UML Process Diagram



UML Deployment Diagram



UML Use Case Diagram  
  
**Diagram

Description automatically generated**

1. **System implementation**Our project was implemented using the Play Framework. We have a frontend project which is responsible for displaying the user interface and communicating with the backend project. The backend project communicates with a MySQL database that we created on our local machines for this project. Our profile page allows for the user to input their first name, last name, occupation/title, a short biography, phone number, and email. Additionally, we have a table that shows the user’s publication history. Here, our goal is for the user to be able to upload their research; if they spoke at a conference, they could include a link to the recording. If they wrote an article, they could provide its title and a link to download the article.
2. **Experiments/Analysis**A user can click on the “Add Publication” button to open a modal with a form. The user can enter the publication’s title, type of media (article, video, etc.), the date it was published, and a link to the content. Upon submitting, the information is sent to our backend.
3. **Known issues, bugs, problems**Unfortunately, we had an issue with our backend that we couldn’t figure out how to fix. Our project was fully connected: the frontend sends POST requests to the backend to be stored in the database, and we confirmed that the field values you input into the form are indeed sent to the backend. Here, we attempt to add a publication with the following values:  
   title = “Test Title”, media\_type = “Article”, date = “2020-04-30”, link = “linktoarticle.com”  
     
   **Text

   Description automatically generated**  
     
     
   You can see at the bottom of the terminal that the backend is printing out the values we sent to it. The frontend does the same:  
   **Text

   Description automatically generated**  
   However, once we send the data to the backend, it does not persist in the database. If you go into the MySQL server instance, no data is being added to the publications table. Although we spent much time on this issue, we could unfortunately not figure it out.  
     
   As a sort of proof-of-concept, we added the “Test Add” button to show what our intended behavior is. This button simply adds some dummy entries into the table, so you can see what we would have liked the “Add Publication” button to accomplish.
4. **Conclusions and future recommended work**We were disappointed that we could not figure out the issue with our database persistence. Even though we did connect the frontend and backend, that is a big issue to have not fixed. However, our work is a good starting point. It took a long time to get comfortable with the Play Framework; it is very different than our previous classwork with React and Node.js. Making our sample profile page definitely taught us the basics of Play. There’s still a lot we would have to learn before our profile page is where we wanted it to be at. However, since the goal was to get comfortable with using Play, we feel like we accomplished that. If we had more time to develop this project, after figuring out the backend issue, we would like to add the following functionality:  
     
    - Ability to display credentials  
    - Ability to manage privacy (make profile private or public)  
    - Ability to apply for account verification  
    - Ability to see recent posts/activity in your community (a dashboard)
5. **Documentation**
   1. **Access Information**

The profile page is located at the /myprofile route. You can customize the route’s parameters to display the information you want. To see a profile page like the one in the screenshots above, use the following link:   
  
<http://localhost:9001/myprofile?&first_name=Landon&last_name=Wood&occupation=CS%20Student&bio=Junior%20Computer%20Science%20Student%20at%20SMU.&phone=214-601-6524&email=landonw@smu.edu>

* 1. **Download and Installation**
     1. First, make sure that you have the Play framework installed properly. If you are using IntelliJ Idea, download the SBT plugin at https://plugins.jetbrains.com/plugin/500   
        7-sbt/versions and install the plugin from disk
     2. Install the .zip from the Github repository: <https://github.com/johnlandonwood/4345Team2Project>
     3. Extract the contents to your chosen location.
     4. Open the frontend and backend projects in separate windows. To do so, find the build.sbt file and open it. IntelliJ should prompt you to “open as project;” choose that option for both frontend and backend.
     5. Start both frontend and backend by clicking the green “Run” icon in the top right.
     6. The project should now be running. Head to the /myprofile link above.
  2. **Assumptions and Considerations**

The above installation instructions assume that you are using the Jetbrains IntelliJ Idea IDE, as that was our preferred IDE for the semester. You can download IntelliJ idea here and following the installation instructions: <https://www.jetbrains.com/idea/>

* 1. **Tutorial**:

Visit the following link to see this page:

<http://localhost:9001/myprofile?&first_name=Landon&last_name=Wood&occupation=CS%20Student&bio=Junior%20Computer%20Science%20Student%20at%20SMU.&phone=214-601-6524&email=landonw@smu.edu>

**Graphical user interface, text, application

Description automatically generated**(You can change any of the route parameters to see what information you want to.)

*Figure 1: My Profile page.*

Upon clicking the “Add Publication” button, the user sees this modal:

**Graphical user interface, text, application, email, Teams

Description automatically generated**

*Figure 2: Add Publication form.*

You can also click on the “Test Add” button to see what the intended behavior of the faulty “Add Publication” button is:

**Graphical user interface, text, application

Description automatically generated**

*Figure 3: Dummy entries in the Publications table.*

This concludes our tutorial and our technical report. Thank you!

**References**

Figure 1: A screenshot of our My Profile page with route parameters already provided.

Figure 2: A screenshot of the form that a user sees when they are trying to add a new publication to their profile page.

Figure 3: A proof-of-concept screenshot of the My profile page with some dummy entries in the Publications table.